DOCUMENT RESUME

ED 076 419 SE 016 063

TITLE	Automated Instructional Management Systems (AIMS) Version III, Program Logic Manual, Volume 2. Source Statement Listings.
INSTITUTION	New York Inst. of Tech., Old Westbury.
SPONS AGENCY	Office of Education (DHEW), Washington, D.C. Bureau of Research.
BUREAU NO	BR-8-0157
PUB DATE	[73]
CONTRACT	OEC-0-8-080157-3691(010)
NOTE	106p.
EDRS PRICE	MF-$0.65 HC-$6.58
DESCRIPTORS	*Computer Assisted Instruction; Computer Programs; *Computer Science; Instruction; *Instructional Media; Instructional Technology; Management Information Systems; Mathematics Education; *Programed Materials
IDENTIFIERS	*Automated Instructional Management System

ABSTRACT

This second program logic manual for the Automated Instructional Management System - Version III contains source statement listings for 23 programs of the system. For related materials, see SE 016 059 through SE 016 064. (DT)

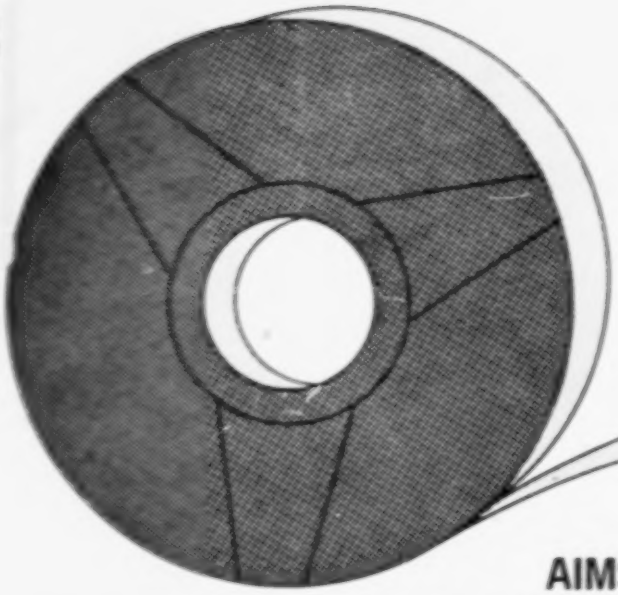
FILMED FROM BEST AVAILABLE COPY

U.S. DEPARTMENT OF HEALTH
EDUCATION & WELFARE
OFFICE OF EDUCATION
THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL OFFICE OF EDUCATION POSITION OR POLICY

ED 076419

ADVANCED SYSTEMS LABORATORY

automated instructional management systems



AIMS VERSION III
PROGRAM LOGIC MANUAL
VOLUME II

NEW YORK INSTITUTE OF TECHNOLOGY
OLD WESTBURY, NEW YORK

SE 016 063

FILMED FROM BEST AVAILABLE COPY

ED 076419

AUTOMATED INSTRUCTIONAL MANAGEMENT SYSTEM

PROGRAM LOGIC MANUAL

SOURCE STATEMENT LISTINGS

Prepared by the Staff of
The Advanced Systems Laboratory

Ernest N. O'Dierno, Director

FILMED FROM BEST AVAILABLE COPY

FOREWORD

- The Automated Instructional Management System (AIMS) was designed to monitor, score, and evaluate individual students, groups of students, and curricular content in a course environment designated for educational management.

- The AIMS System was designed around IBM System/360, and Version III was generated with Model 30/Release 20 IBM Disk Operating System (DOS).

- All source statement listings contained in this manual have been developed with U. S. Office of Eduction funds under Research Contract No. OEC-0-8-080157-3691(010).

ADVANCED SYSTEMS LABORATORY
New York Institute of Technology
Old Westbury, L.I., New York

FILMED FROM BEST AVAILABLE COPY

TABLE OF CONTENTS

	Page
CTIOMAIN	
Markpt	2
Errhed	4
Marksg	5
Function Look	7
Cards	8
Merge	10
Errer	13
Double	16
Single	16
Function Number	18
Function Letter	18
UTILITY ROUTINES	
Info	19
Pack	19
Card In	19
Shifts	20
Lnkedt	21
AHEWMAIN	
Monitor	22
UTILITY ROUTINES	
Image	24
Error	24
This	26
Info	26
Lstcde	27
AIMSHEAD,*	
Header	27
Ctio	36
AIMSCLAS, AIMSHEAD	
Input (Indev)	36

FILMED FROM BEST AVAILABLE COPY

TABLE OF CONTENTS (continued)

	Page
AIMSQUST, AIMSHEAD	
MBO1	41
AIMSPRC1, AIMSHEAD	
Proces(Lesson)	48
AIMSPRC2,*	
Proc1	48
UTILITY PROGRAM	
Ipass	53
AIMSTAPE,*	
Tapsyc	54
AIMSPRC3, AIMSPRC2	
Proc2	57
Function Iround(E)	62
AIMSLIST, AIMSHEAD	
RList (Lesson)	62
UTILITY PROGRAM	
Lnkedt	65
REPTMAIN	
Output Generator	65
HeadPg(Irpt,Ipge)	67
Getit	68
Submit	69

FILMED FROM BEST AVAILABLE COPY

TABLE OF CONTENTS (continued)

	Page
UTILITY PROGRAM	
Ipass	70
REPT0001,*	
Repl	70
REPT0004, REPT001	
Rep04	73
Rep05	76
Rep06	77
REPT0007, REPT0001	
Rep07	79
Decide	79
Divide	82
Conker	84
REPT0012, REPT0001	
Repll	87
Repl2	88
REPT0013, REPT0001	
Repl4	89
Cumave(NF)	90
Getrec	91
Grade	93
Function Sumgrd	94
REPT0015, REPT0001	
Repl5	94
Cumol5	97
Gradl	98
Repl6	99
UTILITY PROGRAM	
Lnkedt	100

FILMED FROM BEST AVAILABLE COPY

```
        OPTS III SOURCE STATEMENT LISTING

// JOB 3041SS       CTIOMAIN
// ASSGN SYSLNK,X'192'
// DLBL IJSYSLN,'SYSLNK',00/365,SD
// EXTENT SYSLNK,000002,1,0,10,1000
// OPTION CATAL
   PHASE CTIOMAIN,*
// EXEC FORTRAN
      DIMENSION ITEST(3)
      DIMENSION ICOM(40)
      INTEGER HEADER,SYSTEM
      INTEGER ICNTRL(15)
      INTEGER*2 LPARM,NPERM,NOS(50),PRS(50),MESREC(65)
      DEFINE FILE 5(1500,33,U,I5),
     1            6( 800,33,U,I6),
     2            7( 200,100,U,I7),
     3          8(4000,35,U,I8),
     4            9( 250,33,U,I9),
     5          10(8040,33,U,I10),
     6            11( 400,23,U,I11),
     7            12( 100,25,U,I12)
      INTEGER*2 IVECT(4,10,40)
      COMMON/CHKREC/IVECT
      COMMON /SYSTEM/ NLESS,NOECK,NREX,NQUEST,NSTUD
      COMMON /FILES/ IFILES(15)
      EQUIVALENCE (IFILES(1),ICD),(IFILES(3),IPT), (IFILES(6),HEADER),
     2(IFILES(12),SYSTEM)
      INTEGER M1/'((MA'/,M2/'NX S'/,M3/'ENSE'/
      DATA IT/'((CA'/,IS/'NDS '/,IME/'MEAD'/,ISTU/'STUD'/,ITP/'NOTA'/
      LOGICAL*1 FIRST
      DO 5 IDEV=1,15
    5 IFILES(IDEV)=IDEV
C
      NLESS=40
      NSTUD= 185
      NREX=4
      NQUEST=48
      NOECK=10
 1000 DO 1 N=1,15
    1 ICNTRL(N)=0
      ICNTRL(6)=0
      ICNTRL(7)=10
      ICNTRL(8)=14
      ICD=1
      IPT=3
      READ (ICD,2) (ITEST(N),N=1,3),ICNTRL(1),ICNTRL(3),ICNTRL(5),ICOM,
     2ICOM
    2 FORMAT (3A4,3X,I2,1X,I1,1X,I2,1X,A4,7X,46A1)
      IF ((ITEST(1).EQ.IT).AND.(ITEST(2).EQ.IS)) GO TO 100
      IF(ITEST(1).EQ.M1.AND.ITEST(2).EQ.M2.AND.ITEST(3).EQ.M3) GO TO 110
    4 WRITE (IPT,3)
      WRITE(IPT,2) (ITEST(N),N=1,3),ICNTRL(1),ICNTRL(3),ICNTRL(5),ICOM,
     2ICOM
    3 FORMAT (1H0,//1H0,'***** ILLEGAL CONTROL CARD *****',/////////////)
      GO TO 1000
  100 IF (ITEST(3).EQ.ISTU) ICNTRL(4)=2
      IF (ITEST(3).EQ.IME) ICNTRL(4)=1
      IF (ICNTRL(4).EQ.0) GO TO 4
      IF (ICNTRL(1).LE.0) GO TO 4
      IF ((ICNTRL(3).NE.4).AND.(ICNTRL(3).NE.5)) GO TO 4
1 PAGE   1
```

FILMED FROM BEST AVAILABLE COPY


```
      ICNTRL(2)=2
      IF (ICON.EQ.(1P)  ICNTRL(2)=1
      GO TO 3000
C  MARK SENSE CONTROL SECTION
  110 IF(ICNTRL(1).EQ.0) GO TO 4
C  PREPOST TEST FORM
      IF(ICNTRL(3).EQ.1) ICNTRL(4)=3
C  STUDY GUIDE FORM
      IF(ICNTRL(3).EQ.2) ICNTRL(4)=4
C  BURN FORMS
      IF(ICNTRL(3).EQ.3) ICNTRL(4)=5
      IF(ICNTRL(4).EQ.0) GO TO 4
      ICNTRL(2)=2
      IF(ICON.EQ.(1P)) ICNTRL(2)=1
 3000 IF (ICNTRL(4).EQ.1) GO TO 4000
      READ (SYSTEM'2) LPERM,NPERM
      READ (SYSTEM'3) PTRS
      READ (SYSTEM'4) NOS
      DO 3003 ICLR=1,40
      DO 3003 ICLS=1,10
      DO 3003 ICLT=1,4
 3003 IVECT(ICLT,ICLS,ICLR)=-1
      DO 3001 ILESS=1,LPERM
      ISTART=PTRS(ILESS)
      IEND=ISTART+NOS(ILESS)-1
      IDIFF=ISTART-1
      DO 3002 ISPOT=ISTART,IEND
      READ (HEADER'ISPOT)HEDREC
      IVECT(1,ISPOT-IDIFF,ILESS)=HEDREC(3)
      IVECT(2,ISPOT-IDIFF,ILESS)=HEDREC(4)
      IVECT(3,ISPOT-IDIFF,ILESS)=HEDREC(10)
      IVECT(4,ISPOT-IDIFF,ILESS)=HEDREC(6)
 3002 CONTINUE
 3001 CONTINUE
C              SO MUCH FOR THAT
 4000 FIRST=.TRUE.
 4001 FORMAT (1H1,//1H0,3A4,3X,I2,1X,I1,1X,I2,1X,A4,7X,40A1,//)
      WRITE (IPT,4001) ITEST,ICNTRL(1),ICNTRL(3),ICNTRL(5),ICON,ICOM
      IDUM = ICNTRL(4)
      GO TO(210,210,220,230,220),IDUM
      GO TO 4
  210 CONTINUE
      CALL CARDS(ICNTRL,FIRST)
      GO TO 250
  220 CONTINUE
      CALL MARKPT(ICNTRL(1),ICNTRL(2))
      IF(ICNTRL(4).EQ.5) GO TO 230
      GO TO 250
  230 CALL MARKSG(ICNTRL(1),ICNTRL(2))
  250 CONTINUE
      CALL EXIT
      END
      SUBROUTINE MARKPT(ICRSE,NOTAPE)
      INTEGER*2 IVECT(4,10,40),JCRSE,JSTUD,CNTREC
      INTEGER*2 ICSN(3),IMINS(2),ISEG(2),IHRS,IVOL(2), ICSE(4),ITYPE
      INTEGER*2 IANS(48),QMRK,QTST/'Q'/,ID(3)/3*'  '/,SEQ/1/
      INTEGER*2 OUTPUT(65),CSN,MINS,SEG,VOL,CSE,TYPE,ANS(48),NUMGOD
      INTEGER*2 NUMGST,TPTEST(5)/'A','C','E','G','I'/,JTP,NUMBAD
      INTEGER*2 CODE(6)/0,2,4,8,16,32/
```

FILMED FROM BEST AVAILABLE COPY


```
      INTEGER*2 TYPCDE(5)/3,5,0,2,1/,NVOL
      COMMON/CHKREC/IVEC(
      COMMON/SYSTEM/NLESS,NDECK,NREX,NQUEST,NSTUD
      COMMON/FILES/IFILES(15)
     0EQUIVALENCE (IFILES(13),MRKINP),(IFILES(3),IOU(),(IFILES(14),INT),
     1(OUTPUT(1),VOL) , (OUTPUT(2),CSN) , (OUTPUT(3),SEG),
     2(OUTPUT(4),TYPE) , (OUTPUT(5),CSE) , (OUTPUT(7),ID(1)),
     3(OUTPUT(12),NUMQST) , (OUTPUT(16),IHRS) , (OUTPUT(17),MINS) ,
     4(OUTPUT(18),ANS(1)) , (OUTPUT(6),SEQ)
      INTEGER*2 MAPANS(48)/1,5,11,16,21,2,7,12,17,22,3,8,13,18,23,4,9,
     114,19,24,5,10,15,20,25,26,31,36,41,46,27,32,37,42,47,28,33,38,43,
     248,29,34,39,44,30,35,40,45/
      CNTREC=0
      JCRSE=ICRSE+1
      JSTUD=NSTUD+1
      NVOL=NLESS+1
      NUMBAD=0
      NUMQST=0
      WRITE(IOUT,610)
  610 FORMAT(1H1,10X,'AIMS MARK SENSE PROCESSING FOR PRE OR POST TEST FO
     1RM (DC6007).')
 1020 CONTINUE
      DO 1030 J=1,5
 1030 OUTPUT(J)=0
      DO 1040 J=10,65
 1040 OUTPUT(J)=0
     0READ(MRKINP,10,END=9999) ICSN(1),IMINS(1),ISEG(1),ICSN(2),IMINS(2)
     1,ISEG(2),ICSN(3),IHRS,IVOL(1),IVOL(2),ICSE(1),ITYPE,ICSE(2),
     2ICSE(3),ICSE(4),(IANS(MAPANS(J)),J=1,48),QMRK
   10 FORMAT(111I,A1,51I1,A1)
      CNTREC=CNTREC+1
      IF(QMRK.EQ.QTST) GO TO 1000
      NUMBAD=NUMBAD+1
      WRITE(IOUT,500) CNTREC
  500 FORMAT('  *** ERROR *** PRE OR POST TEST FORM (DC6007) HAS BAD QUA
     1LITY. RECORD IGNORED.   RECORD NO.. ',I4,/,16X,'CHECK FOR ERASORS,
     2SMEARS,AND DOUBLE MARKS',//)
      GO TO 1020
 1000 CONTINUE
C CONVERT TEST ID DATA
      CSN = ICSN(3)*100 + ICSN(2)*10 + ICSN(1)
      MINS = IMINS(2)*10 + IMINS(1)
      SEG  = ISEG(2)*10 + ISEG(1)
      VOL  = IVOL(2)*10 + IVOL(1)
      CSE  = ICSE(4)*1000 + ICSE(3)*100 + ICSE(2)*10 + ICSE(1)
      DO 1050 JTP=1,5
      IF(ITYPE.EQ.TPTEST(JTP)) TYPE=TYPCDE(JTP)
 1050 CONTINUE
C TEST  ID DATA FOR VALIDITY
      IF(CSN.GT.0.AND.CSN.LE.JSTUD) GO TO 2000
      NUMBAD=NUMBAD+1
      CALL ERRHED(OUTPUT,ICRSE,CNTREC)
      WRITE(IOUT,510)
  510 FORMAT(' COURSE STUDENT NUMBER IN ERROR. ABOVE WORK REJECTED.',/)
      GO TO 1020
 2000 CONTINUE
      IF(VOL.GT.0.AND.VOL.LE.NVOL) GO TO 2020
      NUMBAD=NUMBAD+1
      CALL ERRHED(OUTPUT,ICRSE,CNTREC)
```

FILMED FROM BEST AVAILABLE COPY


```
      WRITE(ICUT,5300) MLES
 5300FORMAT(' VOLUME NUMBER TOO LARGE, LIMIT SET AT, ',I3,'. ABOVE WORK
     1REJECTED.',/)
      GO TO 1020
 2020 CONTINUE
      DO 1060 JCHECK=1,10
      IF(IVEC(1,JCHECK,VOL).NE.SEG) GO TO 1060
      IF(IVEC(2,JCHECK,VOL).NE.ITYPE) GO TO 1060
      GO TO 2010
 1060 CONTINUE
      NUMBAD=NUMBAD+1
      CALL ERRREC(OUTPUT,ICRSE,CNTREC)
      WRITE(ICUT,520)
  520FORMAT(' VOLUME,SEGMENT,OR TYPE OF WORK NOT PRESENT IN AIMS FILES'
     1/,' PLEASE CHECK FILES. ABOVE WORK REJECTED.',/)
      GO TO 1020
C GOOD RECORDS MAKE IT TO HERE
 2010 DO 3000 JANS=1,48
      IF(IANS(JANS).NE.0) NUMQST=NUMQST+1
      ANS(JANS) = CODE(IANS(JANS)+1)+1
 3000 CONTINUE
      IF(NOTAPE.EQ.1) GO TO 3010
      WRITE(IAT) OUTPUT
 3010 GO TO 1020
 9999 CONTINUE
      NUMGOD= CNTREC-NUMBAD
      WRITE(ICUT,600) NUMGOD,NUMBAD
 6000FORMAT(1H0,10X,' AIMS MARK SENSE PROCESSING FOR PRE OR POST TEST(0
     1OR07) COMPLETE.',/,20X'NUMBER OF RECORDS ACCEPTED, ',I4,
     2                 /,20X'NUMBER OF RECORDS REJECTED, ',I4  )
      RETURN
      END
      SUBROUTINE ERRREC(RECD  ,ICRES,RECCNT)
      COMMON/FILES/ IFLE(2),IPT
      INTEGER*2 RECD(17)  ,ICRES,SEQ,JTYPE,INFORM(12),RECCNT
      JTYPE=2
      SEQ = 0
  102 CALL INFO(INFORM)
      WRITE (IPT,103) INFORM,RECCNT  ,SEQ,RECD(6),JTYPE
  103 FORMAT (1H0,/1H ,'**** CARD IN ERROR **** JOB NAME ',4A2,5X,'DATE-
     2 ',4A2,5X,'TIME- ',4A2,'  **********PHYSICAL RECORD - ',I8,/1H ,
     3'PREVIOUS SEQUENCE NO.= ',I2,' CARD SEQUENCE NO.= ',I2,'  TYPE OF
     4 CARDS = ',I1,' (1=HEADER,2=STUDENT     ) ' )
      WRITE (IPT,117) ICRES   ,RECD(5),RECD(1),RECD(3),RECD(4)
  117 FORMAT (1H ,'COURSE BEING PROCESSED = ',I2,' COURSE NUMBER ON CARD
     2 = ',I2,' LESSON = ',I2,' SEGMENT = ',I2,' TYPE = ',I2)
      IF (JTYPE.EQ.1) WRITE (IPT,118) RECD(13),RECD(11),(RECD(N),N=7,9),
     2RECD(10)
  118 FORMAT (1H ,'NUMBER OF QUESTIONS = ',I2,' NUMBER OF SELECTIONS = '
     2,I3,' I.D.FIELD = ',3A2,' (S) OR (D) = 'A1)
      IF (JTYPE.EQ.2) WRITE (IPT,119) RECD(2),(RECD(N),N=7,9),RECD(14),
     2RECD(15),RECD(13),RECD(16),RECD(17)
  119 FORMAT (1H ,'COURSE STUDENT NUMBER = ',I3,' STUDENT I.D. NUMBER =
     2 '3A2,' DATE ON CARD = ',I2,'/',I2,'/',I2,' TIME ON CARD = ',I2,
     3'.',I2)
      WRITE (IPT,121)
  121 FORMAT (1H ,'------------------ E R R O R S  ------------------
     2----------------------------------------------------',/)
      RETURN
```

FILMED FROM BEST AVAILABLE COPY


```
      END
      SUBROUTINE MARKSG(ICASE,NUTAPE)
C     THIS PROGRAM WILL READ SCANNER TAPE AND UNSCRAMBLE PERM 3L6900
      INTEGER*2 IN(121),OUT(65),BLANK,ITYPE(6),IYPE(6),MARK(5),PP(65),
     1RESP,CORF,ERR
      INTEGER*2 IWHOT(4,10,40),BLANK2/'  '/
      INTEGER*2 NUMGOO,NUMBAD,JSTUD,JLESS
      COMMON/SYSTEM/NLESS,NDECK,NREX,NGUESS,NSTUD
      COMMON/CHKREC/IREC
      DATA BLANK/' '/,ITYPE/'A','C','E','G','I','K'/,MARK/3,5,7,17,33/,
     1 IYPE1/3,5,7,2,1,6/
      JSTUD=NSTUD+1
      JLESS=NLESS+1
      NUMGOO=0
      NUMBAD=0
      I3 = 4
      I4 = 14
      OUT(6) = 1
      OUT(13)=0
      OUT(14)=0
      OUT(15)=0
      OUT(7) = BLANK2
      OUT(8) = BLANK2
      OUT(9) = BLANK2
      OUT(10)=0
      OUT(11)=0
      IREC = 0
      WRITE(3,600)
  600FORMAT(1H1,10A,'AIMS MARK SENSE PROCESSING FOR STUDY GUIDE FORM
     1(2C6006).')
C
C---- READ SCANNER TAPE
C
   23 READ(13,1,END=20)(IN(I),I=1,24),(IN(J),IN(J+1),IN(J+2),IN(J+3),
     A IN(J+46),IN(J+49),IN(J+50),IN(J+51),J=25,69,4),IN(121)
     1 FORMAT(32A1,6A,6A1,6A,61A1)
       IREC = IREC + 1
       IERR = 0
       IF(IN(121).EQ.IYPE(6)) 70,71,70
   70 WRITE (3,10) IREC
   10 FORMAT (' RECORD ',I3,' HAS BAD QUALITY--RECORD IGNORED '/)
       GO TO 23
C
C---- CHECK FOR BLANK STUDENT COURSE NUMBER FIELD
C
   71      IF(LOGX(IN(61))* LOGX(IN(64))* LOGX(IN(67))) 21,24,21
   21 WRITE (3,2) IREC
    2 FORMAT (' RECORD ',I3,' HAS A BLANK IN STUDENT COURSE NUMBER FIELD
     1' )
       IERR = IERR + 1
       GO TO 123
   24 JUT (2) =  (4032*IN(67) ) / 256 *100
     1          +(4032*IN(64) ) / 256 *10
     2          +(4032*IN(61) ) / 256
       IF(OUT(2).LE.JSTUD) GO TO 123
       IERR=IERR+1
       WRITE(3,540) IREC
  540FORMAT(' RECORD ',I3,' COURSE STUDENT NUMBER TOO LARGE--RECORD IGN
     1ORED '/)
```

FILMED FROM BEST AVAILABLE COPY


```
C
C---- CHECK FOR BLANKS IN COMPLETION TYPE FIELD
C
  123 IF(LOOK(IN(2)) + LOOK(IN(5)) + LOOK(IN(4)) ) 26,28,26
   26 WRITE (3,3) IREC
    3 FORMAT (' RECORD ',I3,' HAS BLANKS IN COMPLETION TYPE FIELD--FIELD
     1SET TO ZERO' )
      OUT(6) = 0
      OUT(7) = 0
      GO TO 29
   28 OUT(6) = (4032*IN(4)) / 256
      OUT(7) = (4032*IN(5)) / 256 * 10
     1          +(4032*IN(2)) / 256
C
C---- CHECK FOR BLANKS IN SEGMENT FIELD
C
   29 IF(LOOK(IN(14))                    ) 30,31,30
   30 WRITE (3,4) IREC
    4 FORMAT (' RECORD ',I3,' HAS BLANKS IN SEGMENT FIELD ' )
      IERR = IERR + 1
      OUT(3) = 0
      GO TO 32
   31 OUT(3) =(4032*IN(14))/256
C
C---- CHECK FOR BLANKS IN VOLUME FIELD
C
   32 IF(LOOK(IN(16))+LOOK(IN(21)) ) 33,34,33
   33 WRITE (3,5) IREC
    5 FORMAT (' RECORD ',I3,' HAS BLANKS IN VOLUME FIELD' )
      IERR = IERR + 1
      GO TO 35
   34 OUT(1)  =  (4032*IN(21))/ 256 * 10
     1          +(4032*IN(16))/ 256
      IF(OUT(1).LE.JLESS) GO TO 35
      IERR=IERR+1
      WRITE(3,550) IREC
  550 FORMAT(' RECORD ',I3,' VOLUME TOO LARGE-- RECORD IGNORED '/)
C
C---- TEST TYPE FIELD
C
   35 OUT(4) = 4
C
C---- TEST COURSE NUMBER FIELD
C
      IF(LOOK(IN(3))+LOOK(IN(6))+LOOK(IN(9))+LOOK(IN(12)))40,41,40
   40 WRITE (3,16) IREC
   16 FORMAT ( ' RECORD ',I3,' HAS COURSE NUMBER TROUBLE ')
      OUT(5) = 21
      GO TO 42
   41 OUT(5) = (4032*IN(12)) / 256 *1000
     1         +(4032*IN(09)) / 256 *100
     1         +(4032*IN(06)) / 256 *10
     3         +(4032*IN(03)) / 256
      DO 1000 JDECK=1,10
      IF( IVEC(1,JDECK,OUT(1)).NE.OUT(3)) GO TO 1000
      IF( IVEC(2,JDECK,OUT(1)).NE.OUT(4)) GO TO 1000
      GO TO 2010
 1000 CONTINUE
      WRITE(3,520) IREC
PAGE  6
```

FILMED FROM BEST AVAILABLE COPY

```
        AIMS III SOURCE STATEMENT LISTING

  520 FORMAT(' RECORD, ',I3,' VOLUME,SEGMENT,OR TYPE OF MARK NOT PRESENT
     1IN AIMS FILES.'/,10X,'PLEASE CHECK FILES. ABOVE MARK REJECTED.')
      GO TO 23
 2010 CONTINUE
C
C---- COUNT NUMBER OF QUESTIONS ANSWERED
C
   62 KOUNT = 0
      OUT(12) = KOUNT
      DO 60 I=25,117,4
      K = I+3
      DO 61 J = I,K
      IF (IN(J)-16448) 60,61,60
   61 CONTINUE
      KOUNT = KOUNT + 1
   60 CONTINUE
      OUT(12) = 24 - KOUNT
C
C            ZERO OUTPUT AND SET RESPONSE
C
      DO 67 I =16,65
   67 OUT(I)=0
      JJ = 17
      DO 62 I =25,117,4
      K = I + 3
      L = 0
      JJ = JJ + 1
      DO 63 K = I,K
      L = L + 1
      IF(IN(K)-16448)64,63,64
   64 OUT(JJ) = OUT(JJ) + 2**L
   63 CONTINUE
      OUT(JJ) = OUT(JJ) + 1
   62 CONTINUE
C
C---- DONE -- WRITE OUT RESPONSE TAPE AND GO TO READ NEW RECORD
C
      IF(IERR) 55,56,55
   55 WRITE (3,8) IREC,IERR
    8 FORMAT (' RECORD ',I3,' HAS ',I3,' UNRECOVERABLE ERRORS--RECORD IG
     1NORED ' //)
      GO TO 23
   56 CONTINUE
      NUMGOD=NUMGOD+1
      IF(NOTAPE.EQ.1) GO TO 23
      WRITE (14) OUT
      GO TO 23
   20 WRITE(3,9)
    9 FORMAT(1H0,10X,'AIMS MARK SENSE PROCESSING FOR STUDY GUIDE FORM
     1(DC-0006) IS COMPLETE.')
      NUMBAD=IREC-NUMGOD
      WRITE(3,5300) NUMGOD,NUMBAD
 5300 FORMAT(20X,'NUMBER OF RECORDS ACCEPTED, ',I4,
     1     /,20X,'NUMBER OF RECORDS REJECTED, ',I4)
      RETURN
      END
      FUNCTION LOOK(ITEST)
      INTEGER*2 ITEST,DATA(10)
      DATA DATA/'0','1','2','3','4','5','6','7','8','9'/
PAGE  7
```

FILMED FROM BEST AVAILABLE COPY


```
      LOOK  = 0
      DO 20 I=1,10
      IF(ITEST-DATA(I)) 20,21,20
   20 CONTINUE
      LOOK = 1
   21 RETURN
      END
      SUBROUTINE CARDS(ICNTRL,FIRST)
      LOGICAL*1 FIRST
      LOGICAL*1 CHARLY
      DIMENSION ICNTRL(15)
      LOGICAL*1 ACTION
      INTEGER RIGHT,LEFT
      INTEGER*2 IVECT(4,10,40)
      INTEGER*2 AREA(80),RECD(45)
      INTEGER*2 COLS(12),NUMCOL(23)
      INTEGER*2 SHIFTS
      INTEGER*2 IOI,I02,I03,I04,I05
      DATA SCOLS/4,10,16,22,28,34,40,46,52,58,64,70/
      DATA NUMCOL/2,6,8,12,14,18,20,24,26,30,32,36,38,42,44,48,50,54,56,
     2SO,62,66,68/
      COMMON /FILES/ IOO,IPC,IPT,IWI,ISCH,HEADER,IOCT,IQUST,IRKOO,ISCORE
     2,IEXI,SYSTEM
      COMMON/CMKREC/IVECT
      IPT=3
      ITAPE=ICNTRL(8)
      ITAPE=14
      LEFT=1
      RIGHT=2
      ICODE=ICNTRL(4)
      IF (FIRST)            ASSIGN 1 TO ILOC
      IF (.NOT.FIRST)               ASSIGN 2000 TO ILOC
      GO TO ILOC,(1,2000)
    1 REWIND ITAPE
      NREX=0
    2 CALL CRLIN(AREA)
      CHARLY=.FALSE.
      ACTION=.FALSE.
      IF ((AREA(1).EQ.768).AND.(AREA(2).EQ.1056))GO TO 999
      IF ((AREA(1).EQ.768).AND.(AREA(2).EQ.768)) GO TO 1000
      IF ((AREA(1).EQ.768).AND.(AREA(2).EQ.2048)) GO TO 1000
      IF ((AREA(1).EQ.2064).AND.(AREA(2).EQ.2064)) GO TO 1000
      GO TO 99999
 1000 DO 1002 N=1,80
      AREA(N)=LETTER(AREA(N))
 1002 CONTINUE
 1001 FORMAT (1H1,/1H1,'***** JOB STACK ERROR **********',/)
      WRITE (IPT,1001)
      WRITE (IPT,1003) AREA
 1003 FORMAT (1H ,'THIS CARD READ IN BINARY- ',80A1,/1H ,
     2 ' REPAIR JOBSTACK AND CANCEL THIS JOB **********',/////
     3///)
      IF (CHARLY) GO TO 20000
      PAUSE
      GO TO 1000
  999 ACTION=.TRUE.
      GO TO 99999
20000 CALL EXIT
C     TO HAND IN AN EBCDIC CARD IMAGE OF A // JOB CARD
```

FILMED FROM BEST AVAILABLE COPY


```
CC*
C*****************************************************************************
C       IF THE SUPERVISOR CAN BE ENTERED TO GIVE IT A CARD IMAGE,
CC      REMOVE THE COMMENTS CHARACTERS FROM THE FOLLOWING CARDS-
C*****************************************************************************
CC*                                                                           *
C       IF ((AREA(1).EQ.708).AND.(AREA(2).EQ.708)) GO TO 3000
CC*                                                                           *
C THAT CARD REPLACES THE ONE AT STATEMENT 9
CC*                                                                           *
C3000 DO 2001 IN=1,80
CC*                                                                           *
C
C       THIS LOOP CONVERTS THE CARD TO EBCDIC
C
CC*                                                                           *
C       AREA(IN)=LETTER(AREA(IN))
CC*                                                                           *
C2001 AREA(IN)=SHIFTS(AREA(IN),2,8)
CC*                                                                           *
C       DO 2002 IN=1,80,2
CC*                                                                           *
C       THIS LOOP PACKS THE CHARACTERS TWO TO THE HALFWORD
CC*                                                                           *
C
CC*                                                                           *
C       INDEX=(IN/2)+1
CC*                                                                           *
C2002 AREA(INDEX)=SHIFTS(AREA(IN),1,8)+AREA(IN+1)
C
C*****************************************************************************
C*****************************************************************************
99999 CONTINUE
C
C
C    THAT ENDS THE SECTION THAT LOOKS FOR THE END
C
C*****************************************************************************
C
5000  CONTINUE
      IF (ACTION) GO TO 10000
      NREX=NREX+1
C      SET THE COUNTERS TO ZERO,AND THE PHYSICAL RECORD NUMBER TO NREX
      RECD(45)=NREX
      RECD(44)=0
      RECD(43)=0
      DO 3 ILCOL=1,80,2
      IF (AREA(ILCOL).NE.0) RECD(44)=RECD(44)+1
C          PERFORM CHECK FOR ILLEGAL COLUMNS
    3 CONTINUE
      DO 4 NLCL=1,23
      NDEX=NUMCOL(NLCL)
C             CONVERT THE NUMERIC COLUMNS NOW.........
      AREA(NDEX)=NUMBER(AREA(NDEX))
      IF (AREA(NDEX).EQ.-1) RECD(43)=RECD(43)+1
C             AND UPDATE THE NUMERIC COLUMN ERROR COUNTER.......
    4 CONTINUE
      DO 5 IQCOL=1,12
C          MOVE THE QUESTION COLUMNS INTO THE CORRECT AREA
```

FILMED FROM BEST AVAILABLE COPY

```
        AIMS III SOURCE STATEMENT LISTING

        NDEX=NCELS(IQCOL)
     2  RECD(17+IQCOL)=AREA(NDEX)
C
C
C
C         AND FORMAT THE COMMON DATA ON THE CARD
        RECD(1)=(AREA(60)*10)+AREA(62)
        RECD(3)=(AREA(66)*10)+AREA(68)
        RECD(4)=(AREA(54)*10)+AREA(56)
        RECD(5)=(AREA(42)*10)+AREA(44)
        RECD(6)=(AREA(48)*10)+AREA(50)
        ID1=LETTER(AREA(72))
        ID2=LETTER(AREA(74))
        ID3=LETTER(AREA(76))
        ID4=LETTER(AREA(78))
        ID5=LETTER(AREA(80))
C          SEE WHAT KIND OF CARDS ARE ENTERRING
        IF (ICODE.EQ.1) GO TO 100
C
C                    IT'S A STUDENT
        RECD( 2) = ((( AREA(2)*10)+AREA(6))*10)+AREA(8)
        RECD(13)=(AREA(24)*10)+AREA(26)
        RECD(14)=(AREA(12)*10)+AREA(14)
        RECD(15)=(AREA(18)*10)+AREA(20)
        RECD(16)=(AREA(30)*10)+AREA(32)
        RECD(17)=(AREA(36)*10)+AREA(38)
        RECD(7)=ID1
        CALL PACK(ID2,ID3,RECD(8))
        CALL PACK(ID4,ID5,RECD(9))
        GO TO 199
C
C         BELOW HERE, IT'S GOTTA BE A HEADER
  100   RECD(2)=0
        RECD(7)=0
        RECD(8)=0
        RECD(9)=0
        CALL PACK(ID1,ID2,RECD(7))
        CALL PACK(ID3,ID4,RECD(8))
        RECD(9)=ID5
        RECD(10)=0
        RECD(12)=0
        RECD(13)=(AREA(24)*10)+AREA(26)
        RECD(11) = ((( AREA(2)*10)+AREA(6))*10)+AREA(8)
        RECD(14)=0
        RECD(15)=0
        RECD(16)=0
        RECD(17)=0
        RECD(10)=LETTER(AREA(80))
C       BUT NOT OVER HERE IT DOESN'T
  199   CALL MERGE(RECD,ICNTRL,FIRST)
  200   CONTINUE
        GO TO 2
10000   CONTINUE
        END FILE ITAPE
        REWIND ITAPE
2000    CONTINUE
        RETURN
        END
        SUBROUTINE MERGE(RECD,ICNTRL,FIRST)
PAGE 10
```

FILMED FROM BEST AVAILABLE COPY


```
C*******************************************************************************
C*******************************************************************************
C      SUBROUTINE MERGE(RECD,ICNTRL,IVECT,FIRST)
C   THIS SUBROUTINE HANDLES THE MERGING OF INPUT CARDS
C      INTO AIMS TAPE RECORD FORMAT.
C    IT ALSO HANDLES THE SINGLE AND DOUBLE DECISION AND FORMATTING.
C     SUBROUTINE ERRER CHECKS FOR ERRORS IN INCOMING RECORDS.
C    SUBROUTINE DOUBLEAND SUBROUTINE SINGLE HANDLE DOUBLE AND SINGLE
C    RECORD FORMATTING.
C
C*******************************************************************************
C*******************************************************************************
      DIMENSION ICNTRL(15)
      INTEGER OUTAP
      INTEGER*2 ISAVE
      INTEGER*2 S,D
      INTEGER*2 DE,LE,T
      INTEGER*2 ILES
      INTEGER*2 RECD(45)
      INTEGER*2 IVECT(4,10,40)
      INTEGER*2 CHKRC(45)
      INTEGER*2 OUTPUT(65)
      INTEGER*2 NRMTY,ITAPE,JTYPE,NORM
      INTEGER*2 INUM
      INTEGER*2 INUM
      COMMON/CHKREC/IVECT
      LOGICAL*1 ACTION
      LOGICAL*1 FIRST
      LOGICAL*1 DELET
      EQUIVALENCE (OUTPUT(12),INUM)
      DATA DE/'DE'/,LE/'LE'/,T/'T'/
      DATA S/'S'/,D/'D'/
 6000 CONTINUE
      IPT=3
      NORM=ICNTRL(3)
      JTYPE=ICNTRL(4)
      ITAPE=ICNTRL(2)
      NRMTY=ICNTRL(5)
      ILES=RECD(1)
      DELET=.FALSE.
      IF (.NOT.FIRST) GO TO 100
C.......... THIS SECTION IS ONLY EXECUTED ONCE FOR ANY CALL TO CITO
      FIRST=.FALSE.
      DO 1 IN=1,65
C    .......THAT CLEARS THE OUTPUT RECORD..........
    1 OUTPUT(IN)=0
C.........
C    THEN SET UP A FEW REALLY FUN PARAMETERS
C                        ................
      CHKRC(5)=ICNTRL(1)
      ISAVE=CHKRC(5)
      CHKRC(3)=ICNTRL(6)
      CHKRC(2)=ICNTRL(7)
      OUTAP=ICNTRL(8)
C     SET THE SEQUENCE NUMBERS EQUAL TO ZERO
      ISEQ=0
C     AND DROP THROUGH TO AN EXECUTION OF THE OTHER JUNK AND SUCH LIKE
  100 CONTINUE
C     THIS SECTION IS ATTEMPTED FOR EVERY CARD
```

FILMED FROM BEST AVAILABLE COPY


```
      CALL ERRER(RECD,CHKRC,OUTPUT,ISEQ,JTYPE,ACTION)
C        IS THE RECORD LEGAL
      IF (.NOT.ACTION) GO TO 77777
      IF (ISEQ.GT.0) GO TO 2003
      IF ((RECD(7).EQ.DE).AND.(RECD(8).EQ.LE).AND.(RECD(9).EQ.I)) DELET=
     2.TRUE.
      IF (DELET) GO TO 2003
      ICNT=0
C     IF IT ISN'T THE FIRST CARD, I ALREADY KNOW ALL ABOUT IT
C      SEE WHAT KIND OF CARDS THEY ARE
      IF  (JTYPE.EQ.1) GO TO 1000
C     1=HEADER, 2=STUDENT
C
C   THIS NEXT SECTION LOOKS FOR THE CARD IN THE AIMS RECORDS
C..........AND HOPEFULLY FINDS IT
C                          -----
      DO 2 IDECK=1,10
      IF (IVECT(1,IDECK,ILES).NE.RECD(3) ) GO TO 2
      IF (IVECT(2,IDECK,ILES).NE.RECD(4) ) GO TO 2
      GO TO 5
    2 CONTINUE
      CALL ERR(RECD,ISEQ,JTYPE,ICNTRL(1))
      WRITE (IPT,3)
    3 FORMAT (1H ,'REQUESTED LESSON,SEGMENT AND TYPE DON''T EXIST IN AIM
     2S LESSON DATA FILES. PLEASE RE-CHECK.',/1HO,' ABOVE CARD REJECTED'
     3,//)
77777 DO 4 IN=1,85
    4 OUTPUT(IN)=0
      ISEQ=0
      GO TO 99999
    5 ITY=IVECT(3,IDECK,ILES)
      IF (ITY.EQ.0) ITY=5
      IF (ITY.EQ.1) ITY=D
      INUM=IVECT(4,IDECK,ILES)
      GO TO 2000
C
C     THAT'S THE SECTION THAT FINDS OUT ALL ABOUT A STUDENT RECORD
C
C   THIS,ON THE OTHER HAND, DISCOVERS WHAT GIVES ABOUT A HEADER CARD
 1000 CONTINUE
C     THIS SECTION MAKES SURE I HAVE DATA ON A HEADER CARD
      IF((RECD(10).NE.S).AND.(RECD(10).NE.D))
     2 CALL ERR(RECD,ISEQ,JTYPE,ICNTRL(1))
      IF((RECD(10).NE.S).AND.(RECD(10).NE.D))WRITE (IPT,1001)
      IF((RECD(10).NE.S).AND.(RECD(10).NE.D)) GO TO 77777
 1001 FORMAT (1H ,' NO  SINGLE OR DOUBLE INFORMATION ON HEADER CARD',/1H
     20,'ABOVE CARD REJECTED.',//)
      ITY=RECD(10)
      INUM=0
      IF ((RECD(10).EQ.S).AND.(RECD(13).LE.48)) INUM=2
      IF ((RECD(10).EQ.S).AND.(RECD(13).LE.24)) INUM=1
      IF ((RECD(10).EQ.D).AND.(RECD(13).LE.48)) INUM=4
      IF ((RECD(10).EQ.D).AND.(RECD(13).LE.36)) INUM=3
      IF ((RECD(10).EQ.D).AND.(RECD(13).LE.24)) INUM=2
      IF ((RECD(10).EQ.D).AND.(RECD(13).LE.12)) INUM=1
      IF(INUM.GT.1) RECD(6)=INUM
      IF (INUM.NE.0) GO TO 2000
C.............THAT'S IT
C DEAR D. S. YOU HAVE MADE A NONO. THE PREVIOUS STATEMENT WAS INSERTED
```

FILMED FROM BEST AVAILABLE COPY


```
C TO CORRECT THE FACT  THAT THERE WAS NO WAY OF TO UPDATING RECD(6)
C  FOR A MULTI-CARD HEADER . IT WOULD BE CONATSNTLY ONE.      A. L.
      CALL ERR(RECD,ISEQ,JTYPE,ICNTRL(1))
      WRITE (IPT,1004)
 1004 FORMAT (1H ,'INVALID OR MISSING DATA FOR THE NUMBER OF RECORDS',//1
     2H0,'ABOVE RECORD REJECTED',//)
      GO TO 77777
C THIS NEXT SECTION SEES WHAT TO DO NEXT
C
 2000 CONTINUE
      INUM=0
      DO 2001 IN=1,17
 2001 OUTPUT(IN)=RECD(IN)
      ISTART=18
      IF (ITY.EQ.S) INCRE=24
      IF (ITY.EQ.D) INCRE=12
      DO 2002 IN=18,65
 2002 OUTPUT(IN)=0
 2003 ICNT=ICNT+1
      IF (DELET) INUM=ICNT
      IF (DELET) GO TO 2006
      IF (ITY.EQ.S) CALL SINGLE(RECD(18)       ,NORM,NRMTY,RECD(4),NUM)
      IF (ITY.EQ.D) CALL DOUBLE(RECD(18)       ,NUM)
      ISTART=18+((ICNT-1)*INCRE)
      IEND=ISTART+INCRE-1
      IF (NUM.EQ.-1) CALL ERR(RECD,ISEQ,JTYPE,ICNTRL(1))
      IF (NUM.EQ.-1) WRITE(IPT,2005)
 2005 FORMAT (1H ,'ERRORS IN QUESTION COLUMNS.',/1H ,' ABOVE CARD REJECT
     2ED',//)
      IF (NUM.EQ.-1) GO TO 77777
      DO 2004 IN=ISTART,IEND
      ION=IN-((ICNT-1)*INCRE)
 2004 OUTPUT(IN)=RECD(ION)
      TNUM=TNUM+NUM
      ISEQ=ISEQ+1
 2006 IF (ICNT.LT.INUM) RETURN
      OUTPUT(12)=TNUM
      IF (ITY.EQ.S) OUTPUT(10)=0
      IF (ITY.EQ.D) OUTPUT(10)=1
      IF (ICNTRL(2).NE.2) GO TO 77777
 6002 CONTINUE
      WRITE (OUTAP) (OUTPUT(INK),INK=1,65)
      WRITE(3,6003)(OUTPUT(IDUM),IDUM=1,65)
 6003 FORMAT(4(20I6,/))
      GO TO 77777
99999 CONTINUE
      CHKRC(5)=ISAVE
 6001 CONTINUE
      RETURN
      END
      SUBROUTINE ERRER(RECD,CHKRC,AREA,SEQ,JTYPE,ACTION)
C     SUBROUTINE ERRER
C     THIS PROGRAM CHECKS FOR ERRORS IN RECORDS
C     BY DENNIS I. SCHNEIDER
C*************************************************************************
      INTEGER SEQ
      INTEGER*2 ICRES
      INTEGER*2 JTYPE
      INTEGER*2 RECD(45),CHKRC(45),AREA(65)          ,INFORM(12)
```

FILMED FROM BEST AVAILABLE COPY


```
      INTEGER*2 DE,LE,T
      INTEGER*2 SP
      LOGICAL*1 CONTRL(45),CNTRL(45)
      LOGICAL*1 DELET
      LOGICAL*1 IMTC,ILCOL,ACTION,P
      LOGICAL*1 FALSE,TRUE
      LOGICAL*1 NUMS
      EQUIVALENCE(CNTRL(1),CONTRL(1))
      COMMON /SYSTEM/ALESS,NDECK,NREX,NQUEST,NSTUD
      DATA SP/' '/
      DATA DE/'DE'/,LE/'LE'/,T/'T'/
      IERR=0
      MIDDLE=0
      ICHES=CHKRC(5)
      IPT=3
      NUMS=.FALSE.
      FALSE=.FALSE.
      TRUE=.TRUE.
      P=.FALSE.
      ILCOL=.FALSE.
      IMTC=.FALSE.
      DELET=.FALSE.
      DO 1 N=1,45
    1 CONTRL(N)=.FALSE.
      IF (RECD(43).NE.0) NUMS=.TRUE.
      IF (RECD(1).LE.0) CONTRL(1)=.TRUE.
      IF (RECD(1).GT.NLESS) CONTRL(1)=.TRUE.
      IF (JTYPE.EQ.1) GO TO 2
C          JTYPE=1 FOR HEADER,2 FOR STUDENT
      IF (RECD(2).GT.NSTUD) CONTRL(2)=.TRUE.
      IF (RECD(2).LE.0) CONTRL(2)=.TRUE.
    2 IF (RECD(3).LE.CHKRC(3)) CONTRL(3)=.TRUE.
C               CHKRC(3)IS LOWER BOUND FOR SEGMENT NUMBER
C               CHKRC(2) IS UPPER BOUND FOR SEGMENT NUMBER
      IF (RECD(4).LE.0) CONTRL(4)=.TRUE.
      IF (RECD(3).GT.CHKRC(2)) CONTRL(3)=.TRUE.
      IF (RECD(5).NE.CHKRC(5)) CONTRL(5)=.TRUE.
      IF (RECD(6).NE.(SEQ+1))  CONTRL(6)=.TRUE.
      IF ((RECD(7).EQ.DE).AND.(RECD(8).EQ.LE).AND.(RECD(9).EQ.1)) DELET=
     2.TRUE.
      IF (DELET) GO TO 4
      IF (JTYPE.NE.2) GO TO 3
      GO TO 4
    3 IF ((RECD(7)+RECD(8)+RECD(9)).EQ.0) CONTRL(9)=.TRUE.
      IF ((JTYPE.EQ.1).AND.(RECD(10).EQ.SP)) CONTRL(10)=.TRUE.
      IF ((JTYPE.EQ.1).AND.((RECD(11).LE.0).OR.(RECD(11).GT.(24*5*NREX))
     2))CONTRL(11)=.TRUE.
      IF ((JTYPE.EQ.1).AND.((RECD(13).LE.0).OR.(RECD(13).GT.(24*NREX)))
     2)CONTRL(13)=.TRUE.
    4 IF (RECD(44).NE.0) ILCOL=.TRUE.
      IF (SEQ.EQ.0) GO TO 100
C                       SECTION BELOW MAKES SURE OF I.D. AREA MATCH
      IF (JTYPE.EQ.1) INDEX=10
      IF (JTYPE.EQ.2) INDEX=9
      DO 5 I=1,5
      IF (RECD(I).NE.AREA(I))  IERR=IERR+1
    5 CONTINUE
      DO 6 I=7,INDEX
      IF (RECD(I).NE.AREA(I))  IERR=IERR+1
```

FILMED FROM BEST AVAILABLE COPY


```
    6 CONTINUE
      IF (IERR.EQ.0)IMTC=.FALSE.
C                        BELOW HERE,ALL IS OUTPUT OR NO OUTPUT
  100 DO 101 I=1,45
      IF (.NOT.CONTRL(I)) GO TO 101
      GO TO 102
  101 CONTINUE
      IF (ILCCL) GO TO 102
      IF (IMTC) GO TO 102
      IF (NUMS) GO TO 102
      GO TO 7000
      ENTRY ERR(RECD, SEQ,JTYPE,ICRES)
      MIDDLE=1
  102 CALL INFO(INFORM)
      WRITE (IPT,103) INFORM,RECD(45),SEQ,RECD(6),JTYPE
  103 FORMAT (1H0,/1H ,'**** CARD IN ERROR **** JOB NAME ',4A2,5X,'DATE-
     2 ',4A2,5X,'TIME- ',4A2,'  **********PHYSICAL RECORD - ',I8,/1H ,
     3'PREVIOUS SEQUENCE NO.= ',I2,' CARD SEQUENCE NO.= ',I2,'  TYPE OF
     4 CARDS = ',I1,' (1=HEADER,2=STUDENT    ) ' )
      WRITE (IPT,117) ICRES   ,RECD(5),RECD(1),RECD(3),RECD(4)
  117 FORMAT (1H ,'COURSE BEING PROCESSED = ',I2,' COURSE NUMBER ON CARD
     2 = ',I2,' LESSON = ',I2,' SEGMENT = ',I2,' TYPE = ',I2)
      IF (JTYPE.EQ.1) WRITE (IPT,118) RECD(13),RECD(11),(RECD(N),N=7,9),
     2RECD(10)
  118 FORMAT (1H ,'NUMBER OF QUESTIONS = ',I2,' NUMBER OF SELECTIONS = '
     2,I3,' I.D.FIELD = ',3A2,' (S) OR (D) = 'A1)
      IF (JTYPE.EQ.2) WRITE (IPT,119) RECD(2),(RECD(N),N=7,9),RECD(14),
     2RECD(15),RECD(13),RECD(16),RECD(17)
  119 FORMAT (1H ,'COURSE STUDENT NUMBER = ',I3,' STUDENT I.D. NUMBER =
     2 '3A2,' DATE ON CARD = ',I2,'/',I2,'/',I2,' TIME ON CARD = ',I2,
     3'.',I2)
      WRITE (IPT,121)
  121 FORMAT (1H ,'------------------- E R R O R S  --------------------
     2----------------------------------------------------',/)
      IF (MIDDLE.EQ.1) RETURN
      IF (DELET) WRITE (IPT,116)
  116 FORMAT (1H ,'THIS RECORD HAS BEEN FOUND TO BE A DLETE RECORD')
      IF (CONTRL(1)) WRITE (IPT,104)
  104 FORMAT (1H ,'LESSON NUMBER MISSING,UNINTELLIGIBLE OR MIS-MATCH.')
      IF (CONTRL(2)) WRITE (IPT,105)
  105 FORMAT (1H ,'COURSE STUDENT NUMBER MISSING,UNRECOGNIZABLE OR EXCEE
     2DS SYSTEM LIMITATION NSTUD.')
      IF (CONTRL(3)) WRITE (IPT,106)
  106 FORMAT (1H ,'SEGMENT NUMBER MISSING,MISPUNCHED OR UNRECOGNIZABLE,O
     2R PREVIOUS CARDS IN RECORD BAD.')
      IF (CONTRL(4)) WRITE (IPT,107)
  107 FORMAT (1H ,'CARD TYPE NUMBER MISSING OR UNRECOGNIZABLE.')
      IF (CONTRL(5)) WRITE (IPT,108)
  108 FORMAT (1H ,'COURSE NUMBER NOT AS PER SYSTEM SPECIFICATIONS')
      IF (CONTRL(6)) WRITE (IPT,109)
  109 FORMAT(1H ,'SEQUENCE NUMBER MISSING OR BAD, OR PREVIOUS CARD(S) I
     2N THIS RECORD BAD.')
      IF (CNTRL(9).AND.(JTYPE.EQ.2)) WRITE (IPT,110)
  110 FORMAT (1H ,'DATA IN I.D. NUMBER AREA BAD OR MISSING-STUDENT RUN')
      IF (CNTRL(10).AND.(JTYPE.EQ.1)) WRITE (IPT,111)
  111 FORMAT (1H ,'DATA IN I.D. NUMBER AREA BAD OR MISSING-HEADER RUN.')
      IF (CNTRL(11)) WRITE (IPT,112)
  112 FORMAT (1H ,'NUMBER OF SELECTIONS NOT AS PER SPECIFICATION.')
      IF (CNTRL(13)) WRITE (IPT,113)
```

FILMED FROM BEST AVAILABLE COPY


```
  113 FORMAT (1H ,'NUMBER OF QUESTIONS  NOT AS PER SPECIFICATIONS.')
      IF (IECOL) WRITE (IPT,114) RECD(44)
  114 FORMAT (1H ,'ILLEGAL USE OF ODD-NUMBERED COLUMNS-',I4,'-TIME(S).')
      IF (IMTC)  WRITE (IPT,115) IERR
  115 FORMAT (1H ,I5,' OCCASIONS IN WHICH CARD''S IDENTIFICATION DATA IS
     2 MIS-MATCHED WITH THE REMAINDER OF THIS RECORD.')
      IF (NUMS) WRITE (IPT,120)       RECD(43)
  120 FORMAT (1H ,'MISPUNCHED NUMERIC COLUMNS FOUND - ',I3, ' TIMES.',/)
      ACTION=.FALSE.
      GO TO 7001
 7000 ACTION=TRUE
 7001 IF ((JTYPE.EQ.2).AND.((RECD(13).LT.0).OR.(RECD(13).GT.99)))P=.TRUE
     2.
      IF ((JTYPE.EQ.2).AND.((RECD(14).LT.0).OR.(RECD(13).GT.12)))P=.TRUE
     2.
      IF ((JTYPE.EQ.2).AND.((RECD(15).LT.0).OR.(RECD(15).GT.31)))P=.TRUE
     2.
      IF ((JTYPE.EQ.2).AND.((RECD(16).LT.0).OR.(RECD(16).GT.99)))P=.TRUE
     2.
      IF ((JTYPE.EQ.2).AND.((RECD(17).LT.0).OR.(RECD(16).GT.59)))P=.TRUE
     2.
      IF (.NOT.P) GO TO 8000
      IF (ACTION) CALL INFO (INFORM)
      IF (ACTION) WRITE (IPT,103) INFORM,RECD(45),SEQ,RECD(6),JTYPE
      IF (P) WRITE (IPT,7999)
 7999 FORMAT (1H ,'ERRORS IN YEAR,MONTH,DAY,HOURS OR MINUTES COLUMNS - 2
C
C     THIS ROUTINE PERFORMS ONE KIND OF DOUBLE CONVERSION
     2RRORS IGNORED.')
 8000 IF (ACTION.AND.P) WRITE (IPT,8001)
 8001 FORMAT (1H0,'**** ABOVE CARD ACCEPTED DESPITE ERRORS. *****',////)
      IF (.NOT.ACTION) WRITE (IPT,8002)
 8002 FORMAT (1H0,'**** ABOVE CARD HAS BEEN REJECTED **** PLEASE NOTE TH
     2AT IN A MULTI-CARD RECORD,THIS WILL CAUSE OTHER ERRORS **',///////)
      IF (ACTION)  GO TO 10000
      DO 8003 IN=1,65
 8003 AREA(IN)=0
      SEQ=0
10000 RETURN
      END
      SUBROUTINE DOUBLE(AREA,NUM)
C    AS FAR AS I'M CONCERNED, TO **/  WITH DIFFERENT KINDS OF CARDS
C     BY DENNIS I. SCHNEIDER
      INTEGER*2 SHIFTS
      INTEGER*2 AREA(12)
      NUM=0
      DO 200 IER=1,12
      IF (SHIFTS(AREA(IER),RIGHT,10).NE.0) NUM=-1
      IF (NUM.EQ.-1) GO TO 999
  200 CONTINUE
      DO 100 IQUES=1,12
      AREA(IQUES)=SHIFTS(AREA(IQUES),LEFT,1)+1
      IF (AREA(IQUES).NE.1) NUM=NUM+1
  100 CONTINUE
  999 RETURN
      END
      SUBROUTINE SINGLE(AREA,NORM,NRMTY,ITYPE,NUM)
C     THIS ROUTINE HANDLES SINGLE CONVERSION
C     BY DENNIS I. SCHNEIDER
```

FILMED FROM BEST AVAILABLE COPY


```
      INTEGER RIGHT,LEFT
      INTEGER*2  NORM,NRMTY,ITYPE
      INTEGER*2 SHIFTS
      INTEGER*2 AREA(24)
      INTEGER*2 SAVE(24)
      INTEGER*2 HICDE,LWCDE
      INTEGER*2 CODE
      LEFT=1
      RIGHT=2
      IPT=3
      NUM=0
      DO 999 IN=1,24
  999 SAVE(IN)=0
      DO 10000 IQUES=1,12
      CODE=AREA(IQUES)
      IF (NORM.EQ.5) GO TO 100
      IF (NORM.EQ.4) GO TO 200
      WRITE (IPT,1)
    1 FORMAT (1H ,'ILLEGAL NORMAL CARD TYPE CODE- SINGLE.',/)
      CALL EXIT
  100 IF (NRMTY.EQ.ITYPE)       GO TO 2000
      GO TO 1000
C       THIS IS THE END OF THE FIVE-CODE NORMAL SECTION
  200 IF (NRMTY.EQ.ITYPE)        GO TO 1000
      GO TO  2000
C      THIS IS THE END OF THE FOUR-CODE NORMAL SECTION
C     A BRANCH TO 1000 INDICATES 5-CODE FORMATTING
C    A BRANCH TO 2000 INDICATES  4-CODE FORMATTING
C*****************************************************************************
C*****************************************************************************
 1000  CONTINUE
C
C       THESE ARE THE SHIFTS TO FORM THE CODE FOR FIVE RESPONSES
C
      HICDE=SHIFTS(CODE,RIGHT,5)
      LWCDE=CODE-SHIFTS(HICDE,LEFT,5)
      HICDE=SHIFTS(HICDE,LEFT,1)+1
      LWCDE=SHIFTS(LWCDE,LEFT,1)+1
C
C*****************************************************************************
      GO TO 3000
C*****************************************************************************
C
C
C*****************************************************************************
C*****************************************************************************
 2000 CONTINUE
C
C         THESE ARE THE SHIFTS TO FORM CODES FOR FOUR RESPONSES
C
      HICDE=SHIFTS(CODE,RIGHT,5)
      LWCDE=CODE-SHIFTS(HICDE,LEFT,5)
      HICDE=SHIFTS(HICDE,RIGHT,1)
      LWCDE=SHIFTS(LWCDE,RIGHT,1)
      HICDE=SHIFTS(HICDE,LEFT,1)+1
      LWCDE=SHIFTS(LWCDE,LEFT,1)+1
C
C*****************************************************************************
      GO TO 3000
```

FILMED FROM BEST AVAILABLE COPY


```
C*****************************************************************************
 3000 CONTINUE
      SAVE(2*IQUES-1)=LWCDE
      SAVE(2*IQUES)=HICDE
10000 CONTINUE
      DO 997 IERR=1,12
      IF (SHIFTS(AREA(IERR),RIGHT,10).NE.0) NUM=-1
      IF (NUM.EQ.-1)          GO TO 998
 997  CONTINUE
      DO 998 IN=1,24
      IF (SAVE(IN).GT.1) NUM=NUM+1
      AREA(IN)=SAVE(IN)
 998  CONTINUE
      RETURN
      END
      FUNCTION NUMBER(ICODE)
C  THIS ROUTINE HAS BEEN CATALOGED IN THE RELOCATABLE LIBRARY
      INTEGER*2 ICODE,NUMBER
      INTEGER*2 IESTR(10)
      DATA IESTR/512,256,128,64,32,16,8,4,2,1/
      IF (ICODE.EQ.0) NUMBER=NUMBER-NUMBER
C  IS IT AN EXCEPTION
      IF (ICODE.EQ.0) GO TO 7000
C     A DROP THRU MEANS NO
      DO 2 N=1,10
      IF (ICODE.EQ.IESTR(N)) GO TO 3
C     .......LOK IT UP IN THE TABLE TO SEE IF IT'S THERE AT ALL
    2 CONTINUE
      NUMBER=-1
C     A DROP THRU MEANS IT'S ABSENT
      GO TO 7000
    3 NUMBER =N-1
C      THIS SECTION FINDS THE NUMERIC EQUIVALENT AND SPLITS
 7000 RETURN
      END
      FUNCTION LETTER(ICODE)
C  THIS ROUTINE HAS BEEN CATALOGED IN THE RELOCATABLE LIBRARY
C    THIS ROUTINE PERFORMS EBCDIC CONVERSIONS
C     BY DENNIS I. SCHNEIDER
      INTEGER*2 ICODE,LETTER,EQUIV(40),EBCDIC(40)
      INTEGER*2 IAST,ISP
      DATA EBCDIC/'A','B','C','D','E','F','G','H','I','J','K','L','M',
     2'N','O','P','Q','R','S','T','U','V','W','X','Y','Z'
     3,'0','1','2','3','4','5','6','7','8','9',
     4'/','*','(','&'/
      DATA EQUIV/2304,2176,2112,2080,2064,2056,2052,2050,
     22049,1280,1152,1088,1056,1040,1032,1028,
     31026,1025,640,576,544,528,520,516,
     4514,513,512,256,128,64,32,16,
     58,4,2,1,768,1058,2066,2048/
      DATA IAST/'*'/,ISP/' '/
C     FIRST,SEE IF IT'S AN EXCEPTION
      IF (ICODE.EQ.0) LETTER=ISP
      IF (ICODE.EQ.0) GO TO 999
      DO 1 N=1,40
C     .....NEXT,IS IT IN THE TABLE.......
      IF (ICODE.EQ.EQUIV(N)) GO TO 100
    1 CONTINUE
C     A DROP THRU TO HERE MEANS IT'S ABSENT FROM THE TABLE
```

FILMED FROM BEST AVAILABLE COPY

```
        AIMS III SOURCE STATEMENT LISTING

      LETTER = LAST
      GO TO 999
  100 LETTER=EBCDIC(N)
C       THIS PART JUST FETCHES THE EQUIVALENT AND SPLITS
  999 RETURN
      END
/*
// ASSGN SYS$LB,X'191'
// EXEC ASSEMBLY
INFO    START
        COME ENTER
        PRINT NOGEN
JOB     DS    2F
DATE    DS    2F
TIME    DS    2F
PAR     DS    1F
        DS    3C
PAT     DC    10A'F0202021462020482020'
WORK    DS    10C
INTIME  DS    1F
ENTER   L     7,0(0,1)
        GETIME STANDARD
        ST    1,INTIME
        MVC   WORK(10),PAT
        ED    WORK(10),INTIME
        MVC   TIME(8),WORK+2
        COMRG
        MVC   DATE(8),0(1)
        MVC   JOB(8),23(1)
        MVC   0(24,7),JOB
        PRINT GEN
        HOME
        END
/*
// EXEC ASSEMBLY
PACK    START 0
        COME  ENTER
A       DS    1H
B       DS    1H
LOCALE  DSECT
FIELD   DS    1H
PACK    CSECT
ENTER   L     3,0(0,1)
        LH    4,0(0,3)
        STH   4,A
        L     3,4(0,1)
        LH    4,0(0,3)
        STH   4,B
        L     3,8(0,1)
        USING LOCALE,3
        LA    4,FIELD
        MVC   0(1,4),A
        MVC   1(1,4),B
        HOME
        END
/*
// EXEC ASSEMBLY
CKDIN   START 0
*  THIS ROUTINE HAS BEEN CATALOGED IN THE RELOCATABLE LIBRARY
PAGE 19
```

FILMED FROM BEST AVAILABLE COPY

```
         ASMS XIA SOURCE STATEMENT LISTING

*     ENTRY MACRO BRANCHES TO *ENTER*
          COMR  ENTER
*
CLCI      CSR   SYSIS,MCRA
*
*       PHYSICAL EXITS SECTION
*
MCRA      CLM   A*22*,AREA1,A*03*,IOU
*
AREA1     DS    6UM
SAVEA     DS    9D
*
ENTER     STM   0,15,SAVEA
          LALP  CCCI
* GET AN A CARD FROM THE READER
          PAIT  CLCI
          L     3,=F*0000*
*     THIS NEXT SECTION MIGHT - POSSIBLY CAUSE LOOPING AS A TELEMETRY
LOOP      LA    0,3
          S     3,=F*0001*
          TM    0,=X*0002*
          LM    0,AREA1(6)
          SRDA  0,0(0)
          SRA   0,2(0)
          SLDA  0,0(0)
          N     0,=X*00000FFF*
          STM   0,AREA1(6)
          BCT   3,LOOP
          LP    0,15,SAVEA
          L     4,0(0,1)
          LA    3,AREA1
*     PLACE THE DATA IN THE PARAMETER AREA
          MVC   DATA(48),0(3)
          LM    0,15,SAVEA
*.................. AND SPLIT
          MCRE
          EOJ
          LTORG
          END
//*
// LKED ASSEMBLY
SHIFTS    START 0
*  THIS ROUTINE HAS BEEN CATALOGED IN THE RELOCATABLE LIBRARY
*   THIS ROUTINE MANDELS ALL SHIFTING OPERATIONS FOR FORTRAN
*         ENTRY MACRO CAUSES BRANCH TO *ENTER*
          COMR  ENTER
*
*
*     PARAMETER STORAGE AREA
*
*
SAVER     DS    9D
DIRECTN   DS    1F
DISTNCE   DS    1F
DATA      DS    4M
*
*     END OF PARAMETER STORAGE AREA
*
ENTER     STM   0,15,SAVER
```

FILMED FROM BEST AVAILABLE COPY

```
        HP/3  III SOURCE STATEMENT LISTING

*   THIS SAVES ALL THE GPR'S,JUST IN CASE.........
*
*
*    THIS SECTION FETCHES THE ENTRY PARAMETERS
*
          L      4,0(0,1)
          LH     5,0(0,4)
          STH    5,DATA
          L      4,4(0,1)
          L      5,0(0,4)
          ST     5,DIRECTN
          L      4,8(0,1)
          L      5,0(0,4)
          ST     5,DISTNCE
*     I HAVE HAVE ALL THE PARAMETERS
*
*      NEXT, I READ IN THE DATA..........
*
          LM     0,DATA
          L      5,DISTNCE
          L      4,DIRECTN
*
*      ..........AND THEN............
*
          C      4,=F'6000'
*
*           IT'S TEST-THE-CODE TIME,FOLKS........
*               CODE = 1 (LEFT) , 2 (RIGHT)
*
          BE     LEFT
*
*   THIS IS THE RIGHT-SHIFT SECTION
*
          SRL    0,0(5)
          B      SPLIT
*
*
*    ......AND THIS WANDERING STRANGER,IS THE LEFT SHIFT SECTION
*
LEFT      SLL    0,0(5)
SPLIT     STH    0,DATA
          LM     0,5,SAVER
          LM     0,DATA
* EXIT MACRO ALLOWS RETURN TO FORTRAN
          MEND
          EQJ
          ENTRY
          END
/*
// EXEC LNKEDT
/*
/&
// JOB JOB1SB       ANDY,MAIN
// ASSGN SYSLNK,X'192'
// DLBL SJSYSLN,'SYSLNK',00/365,SD
// EXTENT SYSLNK,000062,1,0,10,1000
// OPTION CATAL
 PHASE ANDYMAIN,*OUT
```

FILMED FROM BEST AVAILABLE COPY


```
// CALL FORTRAN
C* AIMS MAIN
      COMMON/SYSTEM/NLESS,NDECK,NREX,NQUEST,NSTUD
         COMMON/FILES/IFIL(15)
         INTEGER DATES(6)
         INTEGER SEGMNT,CONTRO(14),PNAME(2)
         INTEGER * 2 COLUMN(40) ,IA   ,  IDENT
         EQUIVALENCE (COLUMN(1),IA),(CARD(1),IC),
     2   (CARD(2),IC),(CARD(3),ID),(CARD(4),IE)
         INTEGER CARD(4)
         DATA SEGMNT/0/,IDENT/'(('/,
     2   CONTRO/'HEAD','STUD','QUES','PROC','HECK','TORE','RINT','-INP',
     3   '-TO-','-LIS','INPU','DROP','ROST','LIST'/
         DATA PNAME/'MAIN','TOR'/
      DEFINE FILE 5(1500,33,U,I5),
     2            6( 800,33,U,I6),
     3            7( 200,100,U,I7),
     4            8(4000,35,U,I8),
     5            9( 250,37,U,I9),
     6           10(8040,33,U,I10),
     7           11( 500,23,U,I11),
     8           12( 100,25,U,I12)
      NLESS = 40
      NDECK = 10
      NREX = 2
      NQUEST = 48
      NSTUD = 185
      DO 7 I = 1,15
    7 IFIL(I) = I
         IN=1
         IOUT = 3
  100    CONTINUE
  101    FORMAT(40A2)
  102    FORMAT(1H1,6X,'***  AIMS  ***'      ,68X,'JOB'3X,2A4,1X,2A4,1X,
     2   2A4/   1HC,6A,40A2)
         READ(IN,101,END=9000) COLUMN
         IF(IA.NE.IDENT) GO TO 100
         CALL INFO(DATES)
         WRITE(IOUT,102) DATES,COLUMN
         CALL IMAGE(COLUMN,CARD)
         IF(ID.NE.CONTRO(1)) GO TO 2000
 1000    CONTINUE
C A HEADER FUNCTION, GET THE SEGMENT IF NOT IN CORE
         IF(SEGMNT.EQ.1) GO TO 1500
         CALL OPSYS('LOAD','AIMSHEAD')
         SEGMNT = 1
 1500    CONTINUE
         IF(IC.NE.CONTRO(5) ) GO TO 1700
         CALL HEADER
         GO TO 100
 1700    CONTINUE
         IF(IC.NE.CONTRO(6) ) GO TO 1900
C STORE TEMPORARY IN PERMANENT
         CALL STORE
         GO TO 100
 1900    CONTINUE
         IF(IC.NE.CONTRO(7) ) GO TO 7000
         CALL HOLIST
         GO TO 100
```

FILMED FROM BEST AVAILABLE COPY


```
 2000     CONTINUE
          IF(IB.NE.CONTRO(2)) GO TO 3000
          IF(SEGMNT.EQ.2)GO TO 2500
          CALL OPSYS('LOAD','AIMSCLAS')
          SEGMNT = 2
 2500     CONTINUE
          IF(IC.NE.CONTRO(13)) GO TO 2700
C  THIS IS FOR A COURSE ROSTER
          CALL LIST
          GO TO 100
 2700     CONTINUE
          IF(ID.EQ.0) ID = 1
          IF(ID.EQ.1.OR.ID.EQ.13) GO TO 2800
          CALL ERROR(PNAME,0,2,ID)
          GO TO 100
 2800     CONTINUE
          IF(IC.NE.CONTRO(11)) GO TO 2900
          CALL INPUT(ID)
          GO TO 100
 2900     CONTINUE
          IF(IC.NE.CONTRO(12)) GO TO 7000
          CALL DROP(ID)
          GO TO 100
 3000     CONTINUE
          IF(IB.NE.CONTRO(3) ) GO TO 4000
C THIS IS A TO FUNCTION
          IF(SEGMNT.EQ.3) GO TO 3500
          CALL OPSYS('LOAD','AIMSQUST')
 3500     CONTINUE
          IF(IC.NE.CONTRO(8) ) GO TO 3700
C ADD + CHAIN MBO INFORMATION
          IF(ID.EQ.0) ID = 1
          IF(ID.EQ.1.OR.ID.EQ.13) GO TO 3600
          CALL ERROR(PNAME,0,2,ID)
          GO TO 100
 3600     CALL MBO1(ID,IE)
          GO TO 100
 3700     CONTINUE
          IF(IC.NE.CONTRO(9) ) GO TO 3900
C LIST THE TO FILE
          CALL MBOLST
          GO TO 100
 3900     CONTINUE
          IF(IC.NE.CONTRO(10) ) GO TO 7000
C LIST THE QUESTION FILE
          CALL QLIST
          GO TO 100
 4000     CONTINUE
          IF(IB.NE.CONTRO(4) ) GO TO 5000
C ITS TO GRADE A LESSON
          CALL OPSYS('LOAD','AIMSPRC1')
          SEGMNT = 4
          CALL PROCES(ID)
          GO TO 100
 5000     CONTINUE
          IF(IB.NE.CONTRO(14) ) GO TO 7000
          IF(SEGMNT.EQ.5) GO TO 5500
          SEGMNT = 5
          CALL OPSYS('LOAD','AIMSLIST')
```

FILMED FROM BEST AVAILABLE COPY


```
 5500     CONTINUE
          CALL RLIST(ID)
          GO TO 100
 7000     CONTINUE
C BAD     CONTROL  CARD
          CALL ERROR(PNAME,0,1)
          GO TO 100
 9000     CONTINUE
          CALL FINISH
          CALL EXIT
          END
/*
// ASSGN SYSSLB,X'191'
// EXEC ASSEMBLY
IMAGE     START
          COME  ENTER
WORK      DC    ID'0'
ENTER     L     3,0(0,1)
          L     4,4(0,1)
          MVC   0(4,4),2(3)
          MVC   4(4,4),10(3)
          CLC   23(2,3),=CL2'  '
          BE    BLANK1
          PACK  WORK+4(4),23(2,3)
          CVB   5,WORK
          BC    X'F',ST1
BLANK1    SR    5,5
ST1       ST    5,8(0,4)
          CLI   29(3),C' '
          BE    BLANK2
          PACK  WORK+4(4),29(1,3)
          CVB   5,WORK
          BC    X'F',ST2
BLANK2    SR    5,5
ST2       ST    5,12(0,4)
          HOME
          END
/*
// EXEC ASSEMBLY
ERROR     START
          ENTRY FINISH
          COME  ENTER
          PRINT NOGEN
ENTER     ST    1,PARLST          SAVE THE ADDRESS OF PARAMETERS
          BC    X'0',LATER        BRANCH PAST OPEN ON 2ND ENTRY
          MVI   ENTER+5,X'F0'     DISABLE OPEN AT MAIN ENTRY
          MVI   F+1,X'F0'         DISABLE OPEN AT SECONDARY ENTRY
          OPENR PRINTS            OPEN FILE
LATER     L     3,PARLST          GET PARAMETER LIST
          L     4,0(0,3)          GET 1ST PARAMETER ADDR
          MVC   NAME(6),0(4)      1ST PAR IS NAME,MOVED TO PRINT LINE
          L     4,4(0,3)          2ND PARAMETER ADDRESS
          L     5,0(0,4)          GET 2ND PARAMETER
          CVD   5,PDEC            MAKE IT DECIMAL
          MVC   LEVL(2),PATL      GET A PATTERN
          ED    LEVL(2),PDEC+7    EDIT
          MVC   LEVEL(1),LEVL+1   PLACE IN PRINT LINE
*                                 IF LEVEL IS POSITIVE WE WILL ABORT
          BC    X'8',A1           TEST IF EDIT RESULT IS 0
```

FILMED FROM BEST AVAILABLE COPY


```
         MVI   BRANCH+1,X'F0'      ENABLE THE BRANCH TO ABORT
         BC    X'F',A2
A1       MVI   BRANCH+1,X'00'      DISABLE THE BRANCH TO ABORT
A2       LA    7,5                 GPR7=5 REGISTER FOR BCT
         LR    8,7                 GPR8=5 COMPARE REGISTER FOR LST TIME
         A     3,=F'4'             SET UP GPR3 FOR PARAMETERS
         LA    6,NUM-7             LOAD GPR6 WITH BEGINNING OF LINE
         MVC   NUM+1(34),NUM       BLANK THE PRINT LINE
LOOP     A     3,=F'4'             INDEX THE PARAMETER LIST
         L     4,0(0,3)            GET THE PARAMETER ADDRESS
         L     5,0(0,4)            GET THE DATA
         CVD   5,DBEC              MAKE IT DECIMAL
         A     6,=F'7'             MOVE ALONG OUTPUT LINE
         MVC   0(7,6),PATE         PLACE PATTERN IN OUTPUT LINE
         LA    1,6(0,6)            SIG DIGIT ADDR IN GPR1
         EDMK  0(7,6),DBEC+5       SET UP THE LINE
         BC    X'A',CHECK          IF POSITIVE COMPLETE LOOP
         S     1,=F'1'             GET ADDR FOR SIGN
         MVI   0(1),C'-'           INSERT SIGN IN LINE
         CR    7,8                 SEE IF FIRST PARAMETER(ERROR NO.)
         BC    X'7',CHECK          IF NOT COMPLETE LOOP
         MVI   FLAG+3,X'01'        SET FLAG FOR DUMP
CHECK    TM    0(3),X'FF'          SEE IF LAST PARAMETER
         BC    X'7',WRITE          IF LAST PARAMETER LEAVE
         BCT   7,LOOP              MAXIMUM OF 5 PARAMETERS
WRITE    CALL  INFO,(DATA)         GET JOB NAME,DATE + TIME
         MVC   NAMJ(8),DATA        PLACE NAME IN PRINT LINE
         MVC   DATE(8),DATA+8      PLACE DATE IN PRINT LINE
         MVC   TIME(8),DATA+16     PLACE TIME IN PRINT LINE
         CNTRL PRINTS,SP,3,1
         PUT   PRINTS              WRITE THE LINE
BRANCH   BC    X'1',ABORT          IF LEVEL=1,THIS BRANCHES TO ABORT
         L     1,PARLST            RESTORE GPR1
         PRINT GEN
         BCMR  14,0                RETURN TO FORTRAN
         PRINT NOGEN
ABORT    PDUMP 1,X'FFFE'           IF LEVEL = 1 THEN ABORT THE RUN
         EOJ
FINISH   SALR  2,0                 LOAD A USING REGISTER
         USING *,2                 INFORM THE COMPILER OF THIS FACT
F        BC    X'0',PAST           IF CALLED EXTERNALLY WILL OPEN
         OPENR PRINTS              PRINTS IF NECESSARY
PAST     MVC   REP(20),MLD         CHANGE THE HEADING
         MVC   NAME(52),NAME-1     PROPOGATE BLANKS
         CALL  INFO,(DATA)         GET JOB NAME,DATE+TIME
         MVC   NAMJ(8),DATA        MOVE NAME INTO PRINT LINE
         MVC   DATE(8),DATA+8      MOVE DATE INTO PRINT LINE
         MVC   TIME(8),DATA+16     MOVE TIME INTO PRINT LINE
         CNTRL PRINTS,SK,1
         PUT   PRINTS              WRITE
         L     3,FLAG              CHECK IF ANY PROGRAMMER ERRORS
         BCT   3,NONE              IF NONE, THEN NO DUMP
         PDUMP 1,X'FFFE'           DUMP IF ANY PROG ERRORS(NEG NOS.)
NONE     CLOSER PRINTS             CLOSE THE FILE
         EOJ                       RETURN TO MONITOR
PRINTS   DTFPR                                                      X
               BLKSIZE=132,                                         X
               CONTROL=YES,                                         X
               DEVADDR=SYSLST,                                      X
```

FILMED FROM BEST AVAILABLE COPY


```
                DEVICE=1403,                                          X
                IOAREA1=LINE,                                         X
                MODNAME=THIS,                                         X
                RECFORM=FIXUNB
PSEC    DS      1D
FLAG    DC      F'2'
PARLST  DS      1F
LINE    DC      CL1' '
        DC      CL10'***  AIMS '
REP     DC      CL20'ERROR  ***  ROUTINE '
NAME    DC      CL6'       '
        DC      CL6' SEV. '
LEVEL   DC      CL1' '
        DC      CL4' NO.'
NUM     DC      CL7'       '
        DC      CL5' JOB '
NAMJ    DC      CL9'         '
DATE    DC      CL9'         '
TIME    DC      CL8'        '
        DC      20CL1' '
MOD     DC      CL20'END OF JOB  ***      '
DATA    DC      6F'0'
PATL    DC      XL2'F020'
LEVL    DC      XL2'FFFF'
PATE    DC       XL7'40402020202120'
        END
/*
/*
// EXEC ASSEMBLY
THIS    PRMOD                                                         X
                SEPASMB=YES,                                          X
                CONTROL=YES,                                          X
                RECFORM=FIXUNB
        END
/*
// EXEC ASSEMBLY
INFO    START
        COME ENTER
        PRINT NOGEN
JOB     DS      2F
DATE    DS      2F
TIME    DS      2F
PAR     DS      1F
        DS      3C
PAT     DC      10X'F020202148202048202D'
WORK    DS      10C
INTIME  DS      1F
ENTER   L       7,0(0,1)
        GETIME STANDARD
        ST      1,INTIME
        MVC     WORK(10),PAT
        ED      WORK(10),INTIME
        MVC     TIME(8),WORK+2
        COMRG
        MVC     DATE(8),0(1)
        MVC     JOB(8),23(1)
        MVC     0(24,7),JOB
        PRINT GEN
        MORE
```

FILMED FROM BEST AVAILABLE COPY


```
          END
/*
// EXEC ASSEMBLY
LSTCDE   START
         COME  ENTER
CHARS    DC    C'A B C D E F G H I J '
LEGAL    DC    X'0000F800'
WORK     DS    1D
         CNOP  0,4
CHECKS   DC    X'00000002'
         DC    X'00000004'
         DC    X'00000008'
         DC    X'00000010'
         DC    X'00000020'
         DC    X'00000040'
         DC    X'00000080'
         DC    X'00000100'
         DC    X'00000200'
         DC    X'00000400'
ENTER    MVI   WORK,C' '
         MVC   WORK+1(7),WORK
         SR    3,3
         SR    4,4
         L     5,0(0,1)
         LA    6,2(0,0)
         LA    7,18(0,0)
         LA    8,CHECKS
LOOP     LH    9,0(0,5)
         N     9,0(0,8)
         BC    X'8',NOT
         LA    10,CHARS(4)
         A     3,=F'2'
         C     3,=F'8'
         BC    X'2',MANY
         LA    11,WORK-2
         AR    11,3
         MVC   0(2,11),0(10)
         BC    X'F',NOT
MANY     MVI   WORK+7,C'*'
         BC    X'F',DII
NOT      A     8,=F'4'
         BXLE  4,6,LOOP
DII      LH    9,0(0,5)
         N     9,LEGAL
         BC    X'8',DO
         MVI   WORK+7,C'X'
DO       TM    1(5),X'01'
         BC    X'01',DONE
         MVI   WORK+7,C'X'
DONE     L     3,4(0,1)
         MVC   0(8,3),WORK
         HOME
         END
/*
 INCLUDE ILFSSQRT
   PHASE AIMSHEAD,*
// EXEC FFORTRAN
      SUBROUTINE HEADER
C
```

FILMED FROM BEST AVAILABLE COPY


```
C   THE PURPOSE OF THE HEADER FILE IS TO PROVIDE THE SYSTEM WITH A
C DESCRIPTION OF THE COURSE IT IS TO HANDLE. THERE MUST BE ONE HEADER
C FOR EACH DECK ( I.E. TEST ) IN THE COURSE. THIS RECORD CONTAINS THE
C NUMBER OF QUESTIONS IN THE TEST , THE CORRECT ANSWERS, THE NO OF
C PERMISSABLE RESPONSES TO EACH QUESTION , THE MAXIMUM NO. OF SELECTIONS
C IN THE TEST , ETC.  IN ADDITION, THIS RECORD CONTAINS POINTERS TO THE
C QUESTION AND DIRECTIORY FILES . THE PTR TO THE QUESTION FILE ALLOWS
C ACCESS TO A GIVEN QUESTION RECORD MERELY BY ADDING THE QUESTION NO.
C -1 TO THE PRT. THE PTR TO THE DIRECTORY FILE DESCRIBES WHICH TC'S
C ARE WITHIN THIS DECK.
C
C   THIS ROUTINE PROVIDES FOR CREATION, ADDITION, UPDATE, AND LISTING
C OF THE HEADER FILE. LOGICALLY THERE ARE 2 FILES , A SCRATCH AND A
C PERMANENT FILE. RECORDS IN THE SCRATCH FILE ARE NOT USED BY THE SYSTEM
C , THEY ARE MERELY STORED THERE. LOGICALLY   RECORDS 1 - 400 (NLESS *
C NDECK ) BELONG TO THE PERMANET FILE WHILE RECORDS 401 TO 800
C ARE THE TEMPORARY FILE. TOGETHER THEY CONSTITUE THE 800 RECORDS
C OF DATA SET REFERENCE NUMBER 6.
C
C   THERE ARE 3 LOGICAL SECTIONS TO THIS ROUTINE.
C 1)  THE FIRST HANDLES RECORDS IN THE TEMPORARY FILE. ITS FIRST SECTION
C PLACES RECORDS IN THE TEMPORARY FILE IN SORTED ORDER, DELETES OR
C REPLACES RECORDS (IN TEMP.). IT MAY ALSO UPDATE THE ANSWERS (THIS IS
C THE ONLY MODIFICATION PERMITTED TO HEADERS IN THE PERMANENT FILE) OF
C RECORDS IN THE PERMANENT FILE. THE SECOND PORTION IS A SEQUENCE CHECK
C WHICH INSURES THAT THERE ARE NO MORE THAN 10 TEST / LESSON AND THAT
C LESSONS IN TEMP ASCEND BY 1 AND START WITH THE FIRST LESSON FOLLOWING
C THOSE IN PERMANENT FILE. IT ALSO BUILDS THE MAP OF THE TEMPORARY FILE
C FOR THE SYSTEM FILE.
C
C 2)  THIS SECTION CHECKS THE ERROR FLAG FROM THE SYSTEM FILE. IF IT IS
C OFF, IT TRANSFERS RECORDS TO THE PERMANENT FILE, INITIALIZES THEIR
C QUESTION RECORDS, AND SETS UP THE QUESTION PINTERS.  IT BUILDS THE
C MAP OF THE PERMANENT FILE FOR THE SYSTEM FILE.
C
C 3)  THIS SECTION LISTS BOTH THE PERMANENT AND TEMPORARY FILES.
C
C   THE SYSTEM FILE WILL CONTAIN THE FOLLOWING MAP OF THE HEADER FILE.
C        RECORD 2
C               WD 1 = HIGHEST LESSON IN PERMANENT
C               WD 2 = NUMBER OF RECORDS IN PERMANENT
C               WD 3 = HIGHEST LESSON IN TEMPORARY
C               WD 4 = NUMBER OF RECORDS IN TEMPORARY
C               WD 5 = ERROR FLAG (LAST EXECUTION OF SECTION 1 LEFT NO
C                      ERRORS IN TEMPORARY,THEN = 0 , ELSE = 1 )
C
C        RECORD 3
C               WD I = RECORD # OF 1ST HEADER FOR LESSON I IN THE
C                      PHYSICAL FILE.
C
C        RECORD 4
C               WD I = NUMBER OF RECORDS IN THE FILE FOR
C                      LESSON I
C
C        RECORD 5
C               WD 1 = NO. OF RECORDS CURRENTLY IN THE QUESTION FILE
C
      COMMON/SYSTEM/NLESS,NDECK,NREX,NQUEST,NSTUD
      COMMON/FILES/IN1(2),IPRINT,IN2(2),IFILE,IN3,IQUEST,IN4(3),ISYS,
```

FILMED FROM BEST AVAILABLE COPY


```
     2   IN5,ITAPE,IN6
      INTEGER CARD
      INTEGER * 2  RECORD(65,100) ,  RCORD2(65) , PTRS(50) , NOS(50)
      INTEGER * 2  LPERM,NPERM,LTEMP, NTEMP, ERRFLG
      INTEGER * 2  SRTMAP(4,400) , DELET(3) , DROP , ZERO , CHECK
     2   , IRECD(4) , QRECNO ,   QSTION(70)
      INTEGER RNAME(2) , EFLAG , RECD , RC2  , SNAME(2)
      DATA RNAME/'HEAD','ER'/, DELET/'DE','LE','T '/,DROP/1/, ZERO/0/
      DATA   QSTION/40*0,30*' '   ,SNAME/'STOR','E'/
      INTEGER * 2 NUMBER(2), LESSON, SEGMNT, TYPE , COURSE , SEQNCE ,
     2   SCORD , SELECT , GRADES , QUESTS , RESPNS(43) , LABEL(2),QLAB(2)
     3   ,QPTR
      INTEGER  HEADNG(4,2) , ANSWER(2,43) , BLANK , RED , ISTART(2)  ,
     2  DATE(6) , DROPPD(2,2), COND(2)
      INTEGER * 2 QREX(70)
      DATA   HEADNG/'PERM','ANEN','T FI','LE','TEMP','ORAR','Y FI','LE'/
      DATA  BLANK/'    '/ , ISTART/1,401/,LABEL/'.','S.'/,QLAB/' S','LO'
     2/, DROPPD/' ',' ','DELE','TED' / ,COND/'A-OK','STOP'/
      EQUIVALENCE  (LESSON ,  RCORD2(1) )  ,  (SEGMNT ,  RCORD2(3)  )
     2 ,  ( TYPE ,  RCORD2(4) )  , ( COURSE ,  RCORD2(5) )
     3 ,  (SEQNCE ,  RCORD2(6) ) , (SCORE,  RCORD2(10) )
     4 ,  (SELECT ,  RCORD2(11) ) , (GRADES,  RCORD2(12) )
     5 ,  (QUESTS ,  RCORD2(13) ) , ( RESPNS(1) ,  RCORD2(15)  )
     6 ,  (QPTR  ,   RCORD2(8) )
C
C FIRST OBTAIN THE HEADER MAP FORM THE SYSTEM FILE
C
          REWIND ITAPE
          READ(ISYS'2) LPERM, NPERM, LTEMP, NTEMP
          READ(ISYS'3) PTRS
          READ(ISYS'4) NOS
C SET ERROR FLAG TO 0
          EFLAG = 0
          NEW   = 0
          NCHECK = NTEMP
C AND READ A BUNCH OF RECORDS(UP TO 100 AT A TIME )
  200     CALL CTIO(RECORD,NUMB,LC1)
          IF(NUMB.LT.1) GO TO 2100
C
C NOW WORK WITH THEM
C
          DO 2000 CARD = 1,NUMB
C IF THE LESSON IS ALREADY IN PERMANENT STOR IT'S MERELY A CHANGE OF
C ANSWERS
          IF (RECORD(1,CARD).GT.LPERM) GO TO 500
C NOW WE MUST FIND THE RECORD IN PERMANENT,SO LOOK IN THE MAP
          LESS = RECORD(1,CARD)
          I1 = PTRS(LESS)
          I2 = NOS(LESS) + I1 -1
          DO 400 NR = I1 , I2
          READ(IFILE'NR)  RCORD2
          DO 300 J = 3,4
          IF( RCORD2(J).NE.RECORD(J,CARD) ) GO TO 400
  300     CONTINUE
C WE'VE FOUND A MATCH CHECK OTHER PARAMETERS
          IF( RCORD2(5).NE.RECORD(5,CARD)) GO TO 350
          IF( RCORD2(6).NE.RECORD(6,CARD))GO TO 350
          IF( RCORD2(10).NE.RECORD(10,CARD)) GO TO 350
          IF( RCORD2(13).NE.RECORD(13,CARD)) GO TO 350
```

FILMED FROM BEST AVAILABLE COPY


```
C ALL CHECKS SO PERFORM UPDATE
          RCORD2(11) = RECORD(11,CARD)
          RCORD2(12) = RECORD(12,CARD)
          J2 = QUESTS + 17
          DO 325 J = 18,J2
          J3 =  J - 18 + QPTR
          READ (IQUEST'J3) QREX
          RCORD2(J) = RECORD(J,CARD)
          QREX(12) =  RCORD2(J)
  325     WRITE(IQUEST'J3) QREX
C WRITE THE NEW RECORD, AND GO WORK WITH NEXT ONE
          WRITE(IFILE'NR)  RCORD2
          GO TO 2000
  350     J =  RCORD2(3)
          K =  RCORD2(4)
          CALL ERROR(RNAME,0,1,NR,LESS,J,K)
          GO TO 2000
  400     CONTINUE
          J = RECORD(3,CARD)
          K = RECORD(4,CARD)
          CALL ERROR(RNAME,0,2,LESS,J,K)
          GO TO 2000
C ITS NOT AN UPDATE OF PERMANENT SO IT MUST EFFECT TEMPORARY
  500     CONTINUE
C SEE IF ITS A DELET RECORD
          DO 550 I =  1,3
          IF(RECORD(I+6,CARD).NE.DELET(I)) GO TO 1000
  550     CONTINUE
C ITS A DELET SO FIND THE RECORD
          LESS = RECORD(1,CARD)
          I1 = PTRS(LESS)
          I2 = NOS(LESS) + I1 - 1
          IF(I1.LT.401.OR.I2.LT.I1) GO TO 900
  600     DO 800 NR = I1 , I2
          READ(IFILE'NR)  RCORD2
          DO 700  I = 3,4
          IF( RCORD2(I).NE. RECORD(I,CARD) ) GO TO 800
  700     CONTINUE
C WE'VE FOUND THE MATCH SO SET THE DELETE FLAG, AND GO TO NEXT RECORD
          DROP = 1
          WRITE(IFILE'NR)  RCORD2,DROP
          NCHECK = NCHECK - 1
          GO TO 2000
  800     CONTINUE
C WE HAVEN'T FOUND THE RECORD
  900     I = RECORD(3,CARD)
          J = RECORD(4,CARD)
          EFLAG = 1
          CALL ERROR(RNAME,0,3,LESS,J,K)
          GO TO 2000
C THE RECORD ISN'T A DELET, SO ITS A REPLACE OR ADD
 1000     CONTINUE
C  CHECK IF THERE ARE ANY QUESTIONS
          IF (RECORD(13,CARD).GE.1 ) GO TO 1100
          LESS = RECORD(1,CARD)
          I= RECORD(3,CARD)
          J= RECORD(4,CARD)
          EFLAG = 1
          CALL ERROR(RNAME,0,4,LESS,I,J)
```

FILMED FROM BEST AVAILABLE COPY


```
            GO TO 2000
 1100       LESS = RECORD(1, CARD)
            I1 = PTRS(LESS)
            I2 = NOS(LESS) + I1 -1
C IF NONE FOR THIS LESSON ITS AN ADD
            IF(I1.LT.401.OR.I2.LT.I1) GO TO 1500
C OTHERWISE SEE IF ITS A REPLACE
            DO 1200 NR = I1 , I2
            READ(IFILE'NR)  RCORD2
            DO 1150 I = 3,4
            IF ( RCORD2(I) .NE. RECORD(I,CARD) ) GO TO 1200
 1150       CONTINUE
C YES ITS A REPLACE
            WRITE(IFILE'NR) (RECORD(I,CARD),I=1,65),ZERO
            GO TO 2000
 1200       CONTINUE
C NO THIS IS AN ADDITION
 1500       CONTINUE
C  SEE IF THERE'S MORE ROOM IN THE FILE
            I = NEW + NCHECK + NPERM
            IF(I. LT. 400 ) GO TO 1600
            I=RECORD(3,CARD)
            J=RECORD(4,CARD)
            EFLAG = 1
            CALL ERROR(RNAME,0,5,LESS,I,J)
            GO TO 2000
 1600       CONTINUE
            NEW = NEW + 1
            SRTMAP(1,NEW) = RECORD(1,CARD)
            SRTMAP(2,NEW) = RECORD(3,CARD)
            SRTMAP(3,NEW) = RECORD(4,CARD)
            SRTMAP(4,NEW) = 0
            WRITE(ITAPE) (RECORD(I,CARD),I = 1,65)
 2000       CONTINUE
            IF (LCI .EQ . 0) GO TO 200
C IF LAST CARD READ,WRITE OLD
C TEMPORARY RECORDS TO SORT TAPE
 2100       IF(NTEMP.LE.0) GO TO 2300
            I1 = 401
            I2 = NTEMP + I1 -1
            DO 2200 I = I1 , I2
            READ(IFILE'I)  RCORD2,CHECK
            IF( CHECK.NE. 0) GO TO 2200
            NEW = NEW + 1
             SRTMAP(1,NEW) =  RCORD2(1)
            SRTMAP(2,NEW) =  RCORD2(3)
            SRTMAP(3,NEW) =  RCORD2(4)
            SRTMAP(4,NEW) = 0
            WRITE(ITAPE) RCORD2,CHECK
 2200       CONTINUE
 2300       IF(NEW.LE.0) GO TO 4200
C BEGIN TO SORT THE RECORDS, IF ANY
            END FILE ITAPE
            REWIND ITAPE
            DO 3000 IPOS = 1, NEW
            DO 2400 I = 1, NEW
            IF(SRTMAP(4,I  ) .NE. 0) GO TO 2400
            IWH = I
            GO TO 2500
```

FILMED FROM BEST AVAILABLE COPY


```
 2400     CONTINUE
          GO TO 3100
 2500     CONTINUE
          IF(IWH.NE. NEW) GO TO 2600
          SRTMAP(4,IWH) = IPOS
          GO TO 3100
 2600     CONTINUE
          J = IWH + 1
          DO 2900 I = J,NEW
          IF(SRTMAP(4,I).NE.0) GO TO 2900
          IF(SRTMAP(1,IWH) - SRTMAP(1,I) ) 2900 , 2700 , 2650
 2700     IF(SRTMAP(2,IWH) - SRTMAP(2,I) ) 2900 , 2750 , 2650
 2750     IF(SRTMAP(3,IWH) - SRTMAP(3,I) ) 2900 , 2900 , 2650
 2650     CONTINUE
          IWH = I
 2900     CONTINUE
          SRTMAP(4,IWH) = IPOS
 3000     CONTINUE
 3100     CONTINUE
C MAKE SURE TAPE IS REWOUND
          REWIND ITAPE
C AND READ RECORDS BACK ON TO THE DISK
          DO 3200 I = 1  , NEW
          J = SRTMAP(4,I) + 400
          READ(ITAPE)  RCORD2
 3200     WRITE(IFILE'J)  RCORD2 , ZERO
C
C WE'VE FINISHED BUILDING AND UPDATING, SO NOW BEGIN SEQUENCE CHECK
C INITIALIZE
C
 3300     LAST = LPERM
          KOUNT = 0
          I1 = LAST + 1
          DO 3350 I = I1 , NLESS
          PTRS(I) = 0
 3350     NOS(I) = 0
          I2 = 401 + NEW - 1
          DO 4000 I = 401, I2
          READ(IFILE'I) IRECD
          IF(KOUNT.EQ.0) GO TO 3800
          IF( IRECD(1) - LAST ) 3400, 3500, 3700
 3400     CONTINUE
          J = IRECD(1)
          CALL ERROR(KNAME,0,-1,1,LAST,J)
          GO TO 3950
 3500     KOUNT = KOUNT + 1
          IF(IRECD(3).NE.LAST3) GO TO 3600
          IF(IRECD(4).NE.LAST4) GO TO 3600
          EFLAG = 1
          J = I - 400
          CALL ERROR(KNAME,0,6,J,LAST,LAST3,LAST4)
 3600     CONTINUE
          IF(KOUNT.LE.NDECK) GO TO 3950
          EFLAG = 1
          CALL ERROR(KNAME,0,7,LAST,LAST3,LAST4,KOUNT)
          GO TO 3950
 3700     CONTINUE
C STORE THE NUMBER OF RECDS IN THE LAST LESSON
          NOS(LAST) = KOUNT
```

FILMED FROM BEST AVAILABLE COPY


```
C CHECK THE LESSON NUMBER
 3900     CONTINUE
          LAST = LAST + 1
          IF( LAST.EQ.IRECD(1)) GO TO 3900
          J = IRECD(1)
          K = 1 - 400
          EFLAG = 1
          CALL ERROR(KNAME,0,9,K,LAST,J)
          LAST = IRECD(1)
 3900     CONTINUE
          PTRS(LAST) = 1
          KOUNT = 1
 3950     LAST3 = IRECD(3)
          LAST4 = IRECD(4)
 4000     CONTINUE
C STORE KOUNT FOR THE LAST DECK
          NOS(LAST) = KOUNT
          LTEMP = LAST
          NTEMP = NEW
          GO TO 4400
 4200     CONTINUE
          LTEMP = 0
          NTEMP = 0
          IF(LPERM.EQ.NLESS) GO TO 4400
          II = LPERM + 1
          DO 4300 I = II , NLESS
          NOS(I) = 0
 4300     PTRS(I) = 0
 4400     CONTINUE
          ERRFLG = EFLAG
          WRITE(ISYS'2) LPERM, NPERM, LTEMP, NTEMP , ERRFLG
          WRITE(ISYS'3) PTRS
          WRITE(ISYS'4) NOS
 4500     CONTINUE
          GO TO 6800
          ENTRY STORE
C THIS SECTION MUST CHECK THE ERROR FLAG ON THE DISK
C IF ON NO ACTION WILL BE TAKEN, IF OFF ALL RECORDS WILL BE MOVED FROM
C TEMPORARY TO PERMANENT
C
          READ(ISYS'2) LPERM, NPERM, LTEMP, NTEMP, ERRFLG
          IF(ERRFLG .EQ. 0) GO TO 5000
          CALL ERROR(SNAME,0,1)
          GO TO 6800
C READ HEADER MAP
 5000     IF(NTEMP.LE.0)GO TO 6600
          READ(ISYS'3) PTRS
          READ(ISYS'4) NOS
C ALSO READ NUMBER OF QUESTION RECORDS
          READ(ISYS'5) QRECNO
          KOUNT = 0
          LAST = LPERM
          RC2 = 401 + NTEMP - 1
          DO 6000 RECD = 401, RC2
C INCREMENT RECORD NUMBER
          NPERM = NPERM + 1
          READ(IFILE'RECD)  RCORD2
C IF FIRST NONE TO STORE IN NOS   1
          IF(KOUNT.EQ.0 ) GO TO 5100
PAGE 33
```

FILMED FROM BEST AVAILABLE COPY


```
C CHECK FOR LESSON CHANGE
          IF(LAST.EQ. RECORD2(1)) GO TO 5300
          NOS(LAST) = KOUNT
 5100     LAST =  RECORD2(1)
          PERS(LAST) = NPERM
          KOUNT = 1
          GO TO 5500
C THIS IS THE SAME LESSON
 5300     CONTINUE
          KOUNT = KOUNT + 1
C NOW WE PLACE THE RECORD IN PERM, BUT FIRST SET UP QUESTION RECORDS
 5500     CONTINUE
          IQ2 =  RECORD2(3)
          QSTION(1) =  RECORD2(1)
          QSTION(2) =  RECORD2(3)
          QSTION(3) =  RECORD2(4)
          QSTION(4) = RECORD2(5)
          DO 5800 I = 1 , IQ2
          II = QRECNO + I
          QSTION(5) = I
          QSTION(12) =  RECORD2(I+17)
          IF( II.EQ. 4000) GO TO 5800
C NO MORE ROOM
          J = NPERM
          K =  RECORD2(1)
          L =  RECORD2(3)
          M =  RECORD2(4)
          CALL ERROR(SHAPE,1,-1,J,K,L,M)
 5800     WRITE(IQUEST'II) QSTION
C SET UP QUESTION PTR
           RECORD2(7) = 0
           RECORD2(8) = QRECNO + 1
          QRECNO = QRECNO + IQ2
          WRITE(IFILE'NPERM)  RECORD2
          DRUP = 1
          WRITE(IFILE'RECD)  RECORD2 , DRUP
 6000     CONTINUE
C COMPLETE MAP FOR LAST HEADER
          NOS(LAST) = KOUNT
C REWRITE MAP, AND FALL THROUGH TO LIST
          LTEMP = 0
          NTEMP = 0
          LPERM = LAST
          ERRFLG = 1
          WRITE(ISYS'2) LPERM, NPERM, LTEMP, NTEMP, ERRFLG
          WRITE(ISYS'3) PTRS
          WRITE(ISYS'4) NOS
          WRITE(ISYS'5) QRECNO
          ENTRY HDLIST
 6600     CONTINUE
          READ(ISYS'2) LPERM, NUMBER(1), LTEMP, NUMBER(2), ERRFLG
          IPAGE = 0
          CALL INPUT(DATE)
          DO 9000 RED = 1 , 2
          IM = 0
          IF(NUMBER(RED).LE.0) GO TO  9000
          II = ISTART(RED)
          I2 = II - 1  + NUMBER(RED)
          DO 8000 IMA = II, I2
```

FILMED FROM BEST AVAILABLE COPY


```
          IF( IM .GT. 0) GO TO 7000
          IPAGE = IPAGE + 1
          WRITE(IPRINT,6999) DATE ,(HEADNG(J,RED),J=1,4),IPAGE
          IM = 4
 7000     CONTINUE
          IM = IM - 1
C READ THE HEADER RECORD
          READ(IFILE'IRX)  RECORD2 , DROP
          DO 7500 I = 1 , QUESTS
 7500     CALL LSTCOE(RESPNS(I),ANSWER(1,I))
          J2 = IRX - II + 1
          J3 = SCRO + 1
          J4 = DROP + 1
          WRITE (IPRINT,7999) COURSE , LESSON , SEGMNT , TYPE ,
     2   SEQNCE , LABEL(SEGNCE) , J2 , QUESTS  , QLAB(J3) , GRADES
     3   , SELECT , (DROPPD(J,J4),J=1,2)
          IF (QUESTS.GT. 12) GO TO 7700
          WRITE(IPRINT,8001) (J,J=1,QUESTS)
          WRITE (IPRINT,8002) ((ANSWER(J,K),J=1,2),K=1,QUESTS)
          WRITE (IPRINT,8003)
          WRITE (IPRINT,8003)
          WRITE (IPRINT,8003)
          WRITE(IPRINT,8003)
          GO TO 8000
 7700     IF (QUESTS.GT.24) GO TO 7800
          WRITE (IPRINT,8001) (J,J=1,12)
          WRITE (IPRINT,8002) ((ANSWER (J,K),J=1,2),K=1,12)
          WRITE (IPRINT,8001) (J,J=13,QUESTS)
          WRITE(IPRINT,8002) ((ANSWER(J,K),J=1,2),K=13, QUESTS)
          WRITE (IPRINT,8003)
          WRITE (IPRINT,8003)
          WRITE(IPRINT,8003)
          GO TO 8000
 7800     IF (QUESTS.GT.36) GO TO 7900
          WRITE (IPRINT,8001) (J,J=1,12)
          WRITE (IPRINT,8002) ((ANSWER (J,K),J=1,2),K=1,12)
          WRITE (IPRINT,8001) (J,J=13,24)
          WRITE (IPRINT,8002) ((ANSWER(J,K),J=1,2),K=13,24)
          WRITE (IPRINT,8001) (J,J=25,QUESTS)
          WRITE (IPRINT,8002) ((ANSWER (J,K),J=1,2),K=25,QUESTS)
          WRITE (IPRINT,8003)
          WRITE(IPRINT,8003)
          GO TO 8000
 7900     CONTINUE
          WRITE (IPRINT,8001) (J,J=1,12)
          WRITE (IPRINT,8002) ((ANSWER(J,K),J=1,2),K=1,12)
          WRITE (IPRINT,8001) (J,J=13,24)
          WRITE (IPRINT,8002) ((ANSWER(J,K),J=1,2),K=13,24)
          WRITE (IPRINT,8001) (J,J=25,36)
          WRITE (IPRINT,8002) ((ANSWER(J,K),J=1,2),K=25,36)
          WRITE (IPRINT,8001) (J,J=37,QUESTS)
          WRITE (IPRINT,8002) ((ANSWER(J,K),J=1,2),K=37,QUESTS)
          WRITE(IPRINT,8003)
 8000     CONTINUE
          IF( IM . GT. 0)  GO TO 8500
          IPAGE = IPAGE + 1
          WRITE (IPRINT,6999) DATE , (HEADNG(J,RED),J=1,4),IPAGE
 8500     CONTINUE
          WRITE (IPRINT,8501) NUMBER(RED)
```

FILMED FROM BEST AVAILABLE COPY


```
      IF( RED.EQ. 1) GO TO 9000
      K = ERRFLG + 1
      WRITE ( IPRINT,8999) COND(K)
 9000 CONTINUE
      RETURN
 8999 FORMAT(1H1  , 7X , '***  ','AIMS COURSE DESCRIPTION' ,
     2 '  ***' ,    50X   , 'JCB' , 2X , 2A4  ,  2X ,2A4  ,
     3   1X  ,  2A4     /   13X ,  4A4   , 61X   , 'PAGE' ,
     4   I4  /  / )
 7999 FORMAT( 10X , 'COURSE' , 1X ,  I2 ,   11X  , 'LESSON', 1X  ,
     2  I2 ,  11X ,  'SEGMENT', 1X   ,  I2  , 10X  , 'TYPE' ,  1X  ,
     3  I1 ,   4X ,  'CONSISTS OF' ,  I2  ,  1X  ,  'CARD'   ,  A2 ,
     4  5X ,  'RECORD' , I4  /
     5  10X,  'THERE ARE',  I3 , 1X , 'QUESTIONS, EACH OF UP TO'   ,
     6  1X,  A2 , 1X  ,'ANSWERS.', I4 , 1X , 'WILL BE GRADED.'  ,
     7  2X , 'THERE ARE' ,  I4  ,  1X ,'POSSIBLE SELECTIONS.'  , 2X ,
     8   2A4   )
 6501 FORMAT( 1I(/)  ,  13X    , 'THERE ARE' , I4   ,  1X,
     2  'RECORDS IN THIS FILE'  )
 8001 FORMAT(3X,11(I2,9X),I2)
 8002 FORMAT(3X,11(2A4,3X),2A4)
 8003 FORMAT(/)
 8999 FORMAT( 13X , 'THE CONDITION IS' , 1X , A4)
      END
      SUBROUTINE CTIC(RECD,NUM,LCI)
      COMMON/FILES/IW1(12),IH , IW2(2)
      INTEGER * 2 RECD(65,100)
      REWIND IH
      LCI = 1
      NUM = 0
       DO 500 I = 1,100
     READ(IH,END=1000) (RECD(L,I),L=1,65)
      NUM = I
  500 CONTINUE
      LCI = 0
 1000 CONTINUE
     REWIND IH
       RETURN
      END
/*
 PHASE AIMSCLAS,AIMSHEAD
// EXEC FFORTRAN
      SUBROUTINE INPUT(INDEV)
C
C
C  THERE ARE 3 SECTIONS OR ENTRY POINTS TO THIS ROUTINE,EACH WITH ITS
C    OWN FUNCTION. THE MAIN PURPOSE OF THIS ROUTINE IS HANDLING OF THE
C    STUDENT BACKGROUND FILE. THIS FILE SERVES AS A MASTER FILE FOR THE
C    STUDENTS, CONTROLLING EACH PROCESS RUN, AND CONTAINING ALL INFORMA-
C    TION PERTAINING TO EACH STUDENT(OTHER THAN HIS GRADES) . THE
C    STUDENT NUMBER(BY WHICH THE SYSTEM REFERS TO A PARTICULAR STUDENT)
C    IS THE RECORD NO. IN THIS FILE. THE 1ST WORD OF THE FIRST RECORD
C    IN THE SYSTEM FILE CONTAIN THE NO. OF RECORDS IN THIS FILE, THE 2ND
C    WORD IS THE N OF STUDENTS(NOT COUNTING DROPS).
C  THE 3 FUNCTIONS OF THIS ROUTINE ARE
C    1) CREATING OR ADDING TO THE FILE
C        IN THIS FUNCTION THE VALUES READ FOR A STUDENT ARE USED TO
C        CALCULATE A CAPABILITY INDEX FOR EACH STUDENT + PLACE IT IN HIS
C        RECORD
```

FILMED FROM BEST AVAILABLE COPY


```
C     2) DROPPING STUDENTS
C          SETTING THE DROP FLAG IN A STUDENT'S RECORD TO 1
C     3) LISTING THE FILE
C          ALL STUDENTS INCLUDING THOSE DROPPED WILL BE LISTED. THERE
C        ARE CURRENTLY 2 VERSIONS OF THIS ROUTINE ONE FOR EACH USER
C     THEY DIFFER IN THE ALGORITHM FOR CALCULATING THE CAPABILITY
C     INDEX, AND THE FORMATS FOR LISTING THE FILE
C
C   THE FOLLOWING COMMONS SET UP THE DATA SET REFERENCE NOS., AND THE
C   SYSTEM PARAMETERS, THE INPUT UNIT IS THE PARAMETER
        COMMON/SYSTEM/NLESS,NDECK,NREX,NQUEST,NSTUD
        COMMON/FILES/IW1(2),IPRINT,IW2(5),IFILE,IW3(2),ISYS,IW4(3)
C   THE NO. OR RECDS PRESENTLY IN THE FILE, THE NO. OF STUDENTS(FROM THE
C    SYSTEM FILE
        INTEGER * 2 NINFIL,NINCRS
C
C   THE FOLLOWING INPUT VALUES ARE EQUIVALENCED TO THE CORRECT POSITIONS
C    IN THE OUTPUT RECORD
C
        INTEGER * 2 NAME(13),IDNO(5),SATM,SATV,AVR,RANK,ALGE,GEOM,TRIG,
       1 ALGI,CALC,PHYS,IQ,READ,COMENT(6), COURSE , CAPIN , RECORD(65)
       2  , CHEM ,NAVRNK , MTHACH , STUDNO  , SECOND(65)
        EQUIVALENCE (COURSE,RECORD(5)) , (NAME(1),RECORD(6))     ,
       1 (IDNO(1),RECORD(19)) , (CAPIN,RECORD(25)) , (SATM,RECORD(29))    ,
       2 (SATV,RECORD(30)) , (AVR,RECORD(31)) , (RANK,RECORD(32))        ,
       3 (ALGE,RECORD(33)) , (GEOM,RECORD(34)) , (TRIG,RECORD(35))       ,
       4 (ALGI,RECORD(36)) , (CALC,RECORD(37)) , (PHYS,RECORD(38))       ,
       5 (IQ,RECORD(39)) , (READ,RECORD(40)) ,  (NAVRNK,RECORD(41))      ,
       6 (COMENT(1),RECORD(42)) , (MTHACH ,RECORD(39) ) ,(CHEM,RECORD(38))
       7     , (STUDNO, RECORD(2) )
        DATA RECORD/41*0 , 14 * ' ' ,  10 * 0 /
        DIMENSION NERR(2)
        DATA NERR/'INPU','T'/
C
C   THESE WORDS ARE FOR THE DROP SECTION
C
        INTEGER DNAM(2)
        DATA DNAM/'DROP',' '/
        INTEGER * 2 DATE(12) , STUDNM(13),DROPFG
        EQUIVALENCE (DROPFG,RECORD(24))
        INTEGER * 2 BLANK
        DATA BLANK/' '/
C
C   LIST SECTION
C
        INTEGER  * 2 DATIN(4),DATOUT(4)
        EQUIVALENCE (DATIN(1) , RECORD(48)) , ( DATOUT(1), RECORD(52) )
        INTEGER DROPD(2)
        DATA DROPD/'  ','YES'/
C   CLEAR RECORD AREA
C   THIS IS THE MAIN ENTRY FOR FILE CREATION
C
C   OBTAIN THE NOS. OF RECORDS AND STUDENTS AND THE DATE
        NOS = 0
        READ(ISYS'1) NINFIL,NINCRS
        CALL INFO(DATE)
C   SND BEGIN TO INPUT STUDENTS
  200 CONTINUE
        DO 10 I = 1,41
```

FILMED FROM BEST AVAILABLE COPY


```
   10 RECORD(I) = 0
      DO 20 I = 42,55
   20 RECORD(I) = BLANK
      DO 30 I = 56,65
   30 RECORD(I) = 0
      READ(INDEV,201,END=2800) NAME,IDNO,SATM,SATV,AVR,RANK,ALGE,GEOM,
     1 TRIG,ALGI,CALC,PHYS,IQ,READ,COMENT,COURSE,STUDNO
  201 FORMAT(12A2,A1,  4A2,A1, I3 , I3 ,  I2 , I1 , I2 , I2 , I2 , I2 ,
     1 I2 , I2  , I3 ,  I2 , 3X  ,  6A2  ,  I2  ,  I3  )
      NOS = NOS + 1
C
C IF STUDNO ISN'T 0 THIS IS AN UPDATE CARD AND REPLACES AN EXISTANT
C   RECORD
C
      IF (STUDNO .EQ. 0) GO TO 299
C THIS IS AN UPDATE SO CHECK THE NUMBER
      IF (STUDNO.GE.1.AND.STUDNO.LE.NINFIL) GO TO 275
      I=STUDNO
      CALL ERROR(NERR,0,1,NOS,I)
      GO TO 200
C THE NO. IS OK SO READ THE OLD RECORD(DATES,DROPPS,ETC. MUST BE SAVED)
  275 READ(IFILE'STUDNO) SECOND
C CHECK THAT THE NAMES MATCH (ONE STUDENT MAY NOT REPLACE ANOTHER)
      DO 280 I = 1,13
      IF (NAME(I).EQ.SECOND(I+5)) GO TO 280
      J=STUDNO
      CALL ERROR(NERR,0,2,NOS,J,I)
      GO TO 200
  280 CONTINUE
C ALL CHECKS SO WE CAN GO CALCULATE THE CAPABILITY INDEX
      GO TO 400
  299 IF(NSTUD - NINFIL) 300, 300 ,400
C  THERE'S NO MORE ROOM, SO LEAVE
  300 CALL ERROR(NERR,0,3,NOS,NSTUD)
      GO TO 2800
C
C  MORE ROOM ,SO PLACE STUDENT NO IN RECORD CALCULATE CAPIN, AND
C    WRITE OUT THE RECORD
C
  400 CONTINUE
C
C          CAPABILITY INDEX CALCULATIONS DEFFERS FOR EACH USER
C  AS NO INPUT VALUE COULD REALISTICALLY BE 0 , 0 INDICATES
C  MISSING DATA
C
C  THIS ONE IS FOR HEW
C
      C1 = 0.
      C2 = 0.
      SUM1=0.
      SUM2=0.
      DO 600 I = 33,37
      IF( RECORD(I). EQ . 0 ) GO TO 600
      SUM1 = SUM1 + RECORD(I)
      C1 = C1 + 1.
  600 CONTINUE
      IF(SATM.EQ.0) GO TO 1000
  800 CONTINUE
      C2 = C2 + 1.
```

FILMED FROM BEST AVAILABLE COPY


```
      SUM2 = SUM2 + (SATM - 200.)/ 6.
1000  CONTINUE
      IF(SATV.EQ.0) GO TO 1200
      C2 = C2 + 1.
      SUM2 = SUM2 + (SATV - 200.)/ 6.
1200  CONTINUE
      IF(AVR.EQ.0) GO TO 1400
      C2 = C2 + 1.
      SUM2 = SUM2 + AVR
1400  CONTINUE
      IF(RANK.EQ.0) GO TO 1600
      C2 = C2 + 1.
      SUM2 = SUM2 + (RANK +1) * 10
1600  CONTINUE
      IF(PHYS.EQ.0) GO TO 1800
      C2 = C2 + 1.
      SUM2 = SUM2 + PHYS
1800  CONTINUE
      IF(IQ.EQ.0) GO TO 2000
      C2 = C2 + 1.
      SUM2 = SUM2 + IQ /2.
2000  CONTINUE
      IF(READ.EQ.0) GO TO 2200
      C2 = C2 + 1.
      SUM2 = SUM2 + READ * 7.
2200  CONTINUE
      IF(C1.EQ.0) GO TO 2400
      C2 = C2 + 1.
      SUM2 = SUM2 +(SUM1/C1)
2400  CONTINUE
      IF ( C2 .GT.0.9 ) GO TO 2500
      CAPIN = -999
      GO TO 2550
2500  CAPIN = SUM2 / C2
2550  CONTINUE
      IF (STUDNO.GT.0) GO TO 2700
      NINCRS = NINCRS + 1
      NINFIL = NINFIL + 1
      STUDNO = NINFIL
      DO 2600 I = 1,4
2600  DATIN(I) = DATE(I+4)
      WRITE(IFILE'NINFIL) RECORD
      GO TO 200
C THIS WAS AN UPDATE
2700  CONTINUE
      RECORD(24) = SECOND(24)
      DO 2710 I =26,28
2710  RECORD(I) = SECOND(I)
      DO 2720 I =48,65
2720  RECORD(I) = SECOND(I)
C HAVING PRESERVED INFORMATION, REWRITE THE RECORD
      WRITE(IFILE'STUDNO) RECORD
      GO TO 200
C
C  HAVING FINISHED, UPDATE THE SYSTEM FILE AND GO LIST
C
2800  CONTINUE
      WRITE(ISYS'1) NINFIL,NINCRS
      GO TO 5000
```

FILMED FROM BEST AVAILABLE COPY


```
C
C  THIS SECTION DELETES STUDENTS FROM THE SYSTEM
C
      ENTRY DROP(INDEV)
 3000 CONTINUE
C  GET SYSTEM COUNTS (NO. OF RECORDS,NO. OF STUDENTS)
      READ(ISYS'1) NINFIL,NINCRS
C  GET CURRENT DATE
      CALL INFO(DATE)
      NOS = 0
C
C  BEGIN READING STUDENTS
C
 3300 READ(INDEV,3301,END=4800) STUDNM,STUDNO
 3301 FORMAT(12A2,A1,  52X  , I3 )
      NOS = NOS +1
C  CHECK STUDNO FOR VALIDITY
      IF ( STUDNO.GT.0 .AND. STUDNO.LE. NINFIL ) GO TO 3600
      J = NINCRS
      I=STUDNO
      CALL ERROR(DNAM,0,1,NOS,I,J)
      GO TO 3300
C  READ THE STUDENT RECORD
 3600 READ(IFILE'STUDNO) RECORD
      IF(DROPFG.EQ.0 ) GO TO 4000
      I = STUDNO
      CALL ERROR(DNAM,0,2,NOS,I)
      GO TO 3300
C  IF STILL IN COURSE CHECK THE NAME
 4000 CONTINUE
      DO 4300 I = 1,13
      IF(STUDNM(I) .EQ. NAME(I) ) GO TO 4300
      J = STUDNO
      CALL ERROR(DNAM,0,3,NOS,J,I)
      GO TO 3300
 4300 CONTINUE
C  ALL CHECKS,DROP STUDENT
      DROPFG = 1
C  PLACE DATE DROPPED IN RECORD
      DO 4600 I = 1,4
      DATOUT(I) = DATE(I+4)
 4600 CONTINUE
      WRITE(IFILE'STUDNO) RECORD
C
C  CORRECT COUNT OF STUDENTS
C
      NINCRS = NINCRS - 1
      GO TO 3300
C  WE'VE DROPPED ALL STUDENTS
 4800 WRITE(ISYS'1) NINFIL,NINCRS
C  AND FALL THROUGH TO LIST ROUTINE
C
C  THE PURPOSE OF THIS SECTION IS TO PRODUCE A COURSE LISTING
      ENTRY LIST
 5000 CONTINUE
C  GET THE NO. OF RECORDS, AND OF STUDENTS, AND DATE,ETC.
      READ (ISYS'1) NINFIL,NINCRS
      CALL INFO(DATE)
      IH = 0
```

FILMED FROM BEST AVAILABLE COPY


```
      IPAGE = 0
      DO 9000 I = 1 , NINFIL
      IF( IH .GT. 0 ) GO TO 8000
      IPAGE = IPAGE + 1
      WRITE(IPRINT,7999) DATE,IPAGE
      WRITE(IPRINT,7998)
      IH = 40
 7999 FORMAT(1H1     ,   7X  , '***'  , 2X , 'AIMS COURSE ROSTER' ,  2X,
     1'***',55X,'JOB', 2X,4A2   ,   2X  ,    4A2  ,  1X   ,  4A2       ,
     2   / 91X,   'PAGE' , I4  /  /   /      )
 7998 FORMAT(
     3 1X,  'NAME' ,  22X , 'S.S. NUM.' ,  1X , 'NO.' , 1X ,  'CAP.'
     4  , 1X ,  'SAT' , 2X , 'SAT' ,  1X , 'AVR' ,  4X , 'ALG' , 2X  ,
     5 'ALG' ,  1X , 'GEOM' , 4X , 'CALC' , 5X , 'IQ' ,4X,'COMMENTS'  ,
     6  5X,    'CRSE',1X  , 'DROP' , 1X  , 'DATE' , 5X , 'DATE'      /
     7 41X , 'INDX' ,  1X  ,'MATH' , 1X , 'VERB' , 3X , 'RNK'      ,
     8  1X , 'ELEM' ,  1X  , 'INTR'  , 4X ,'TRIG' ,  4X  ,'PHYS'
     9  3X ,'READ' ,  23X  , 'ENTERED' ,  2X  , 'DROPPED'  /  /    )
 8000 READ(IFILE'I) RECORD
      IH = IH -1
      K = DROPFG + 1
      WRITE(IPRINT,8001) NAME , IDNO, STUDNO , CAPIN,SATM,SATV,AVR,RANK,
     1 ALGE, ALGI, GEOM , TRIG , CALC , PHYS , IQ , READ,
     2 COMENT , COURSE ,DROPD(K) , DATIN ,DATOUT
 8001 FORMAT( 1X , 12A2,A1 ,  1X ,  4A2,A1 ,   1X , I3  ,   2X ,  I2 ,
     2  2X , I3  ,  2X ,  I3 ,  2X , I2  ,  1X , I1  ,  3X , I2  ,
     3  3X ,  I2  ,    2X , I2  , 2X  I2  ,    2X ,  I2  ,  2X , I2  ,
     4  1X ,  I3 ,   1X ,  I2 ,   1X , 6A2   ,  2X  ,  I2  ,  2X,  A3 ,
     5  1X ,  4A2  ,   1X  , 4A2     )
 9000 CONTINUE
      IPAGE = IPAGE+1
      WRITE(IPRINT,7999) DATE,IPAGE
      WRITE(IPRINT,9001) NINFIL,NINCRS
 9001 FORMAT(1X, 9(/)                 ,
     1  13 X   , 'THERE ARE' ,I4  ,' RECORDS IN THE STUDENT FILE'     /
     2  14 X   , 'OF THESE'  ,I4  ,' STUDENTS REMAIN IN THE COURSE '  )
      RETURN
      END
/*
 PHASE AIMSQUST,AIMSHEAD
// EXEC FFORTRAN
        SUBROUTINE MBOI(INUNIT,LEVEL)
C
C     THE PURPOSE OF THIS ROUTINE IS TO ADD MBO INFORMATION TO THE
C QUESTION FILE AND THEN CHAIN THE INFORMATION IN SUCH A WAY THAT
C ALL QUESTIONS RELATED TO THE SAME TO  THIS ROUTINE READS THE SYSTEM
C FILE ONLY TO ACCESS THE HEADER FILE, IT READS THE HEADERS FROM THE
C PERMANENT FILE, IN ORDER TO BUILD A MAP OF THE QUESTION FILE IN CORE.
C THIS MAP ALLOWS A DIRECT ACCESS OF ANY QUESTION RECORD. THUS MBO CARDS
C MAY BE IN ANY ORDER, THOUGH TIME WILL BE SAVED IF THEY ARE IN LESSON,
C SEGMENT + TYPE ORDER.
C
C     THE FIRST SECTION OF THIS ROUTINE BUILDS THE TABLE, READS THE MBO
C CARD, READS THE CORRECT QUESTION RECORD , ADDS THE MBO INFORMATION,
C AND REWRITES THE RECORD.  THIS PREVENTS DESTRUCTION OF ANY QUEST.
C VALIDITY INFO WHICH MAY BE IN THE RECORD FROM A PROCESS RUN.
C
C     THE SECOND SECTION DETERMINES WHERE TO START CHAINING FROM, AND
C THEN CHAINS FROM THAT POINT.  THIS PREVENTS UN-NECESSARY RECHAINING,
```

FILMED FROM BEST AVAILABLE COPY


```
C YET ALLOWS THE SAME ROUTINE TO BE USED FOR INITIAL CHAINING, AND
C SUBSEQUENT UPDATE.
C
C     THE FILES USED BY THIS ROUTINE ARE AS FOLLOWS
C
C     SYSTEM FILE
C 1) RECORDS 2,3, +4 ARE READ IN ORDER TO FACILITATE READING OF THE
C HEADER FILE. RECORD 5 WILL BE READ TO DETERMINE HOW MANY RECORDS
C ARE IN THE QUESTION FILE.
C RECORD 6 WILL CONTAIN THE NUMBER OF DIRECTORY RECORDS CURRENTLY USED.
C     HEADER FILE
C 1) THE HEADER FILE POINTERS ARE READ INTO CORE IN ORDER TO BUILD A
C TABLE WHICH ALLOWS DIRECT ACCESS OF THE QUESTION FILE
C 2) THE DIRECTORY POINTERS ARE CHECKED TO DETERMINE WHERE WE LAST CHAIN
C -ED, AND AS THE DIRECTORY RECORDS ARE CREATED, THEIR NUMBERS ARE
C PLACED IN THE DIRECTORY POINTERS OF THE CORRECT HEADER RECORDS.
C
C
C     QUESTION FILE
C 1) THE QUESTION RECORDS FOR WHICH MBO INFORMATION IS READ ARE
C READ, THE INFO. ADDED, AND ARE THEN REWRITTEN. THUS QUESTION RECORDS
C ARE CREATED IN SUBROUTINE HEADER. THIS ROUTINE ADDS MBO INFORMATION.
C THE PROCESS ROUTINES WILL ADD THE QUESTION VALIDITY INFORMATION
C 2) A PASS WILL BE MADE THROUGH THE QUESTION FILE TO READ THE MBO INFO,
C AND PLACE IT IN A TABLE. CHAINING WILL TAKE PLACE IN THE TABLE.
C A FINAL PASS WILL READ + REWRITE EACH RECORD WITH THE CORRECT CHAIN
C POINTERS.
C
C     DIRECTORY FILE
C 1) THE DIRECTORY FILE WILL CONTAIN A RECORD FOR EACH SCOPE(A SCOPE IS
C THE AREA WHICH THE DEFINITION OF THE TO ENCOMPASSES. A SCOPE WILL BE
C EITHER A LESSON , OR A SEGMENT.  WORD I OF EACH RECORD WILL POINT
C TO THE FIRST RECORD OF THE CHAIN FOR TO NO. I. THUS A DIRECTORY RECORD
C IS EQUIVALENT TO A SCOPE. A SCOPE WILL BE A LESSON IF LEVEL = 0, OTHER
C -WISE IT WILL BE THE ENTIRE LESSON.
C
C     THE FIRST PARAMETER, INUNIT SPECIFIES THE DEVICE FROM WHICH THE
C MBO RECORDS(80-CHAR.,EBCDIC) WILL BE READ.
C THE FIRST DIMENSION WILL INDEX BY LESSON , THE SECOND BY DECK NO.,
C THE THIRD WILL BE 1-4, WHERE WORD 1 CONTAINS THE SEGMENT NO.,
C             WORD 1 = SEGMENT NO.
C             WORD 2 = TYPE
C             WORD 3 = RECORD NO. OF FIRST QUESTION RECORD
C             WORD 4 = NUMBER OF QUESTION RECORDS
C
      COMMON/SYSTEM/NLESS,NDECK,NRES,NQUEST,NSTUD
      COMMON/FILES/IW1(2),IPRINT,IW2(2),IHEAD,IDIR,IQUEST,IW3(3)
     2   ,ISYS,IW4(3)
      INTEGER * 2  LPERM  ,  NPERM  , IOREC ,
     2   PTRS(50),     NOS(50)  , MRCORD(65) , LTABLE(4,10,40)
     3   , MRCORD(40)  ,LESSON,  SEGMNT , TO , MBO ,KEY(17)  ,
     4   SKILL, SKIL2 , MEDIA, TYPE, QUEST , PRES(13) ,COURSE
      INTEGER * 2 QSTION(70) , DRCTRY(200), CHAIN(3,4000)
     2   , POINT1 , POINT2  , QPOINT    , QTO  ,   QMBO
      INTEGER RNO , DECK , RNAME(2) , HEAD , POINT
      DATA RNAME/'MBO',' '/
      EQUIVALENCE   (LESSON  , MRCORD(1) )  , (SEGMNT , MRCORD(2) ) ,
     2  (TO , MRCORD(3) ) ,  (MBO , MRCORD(4)   ), (KEY(1) , MRCORD(5))
     3  , (SKILL, MRCORD(22) ) ,   (SKIL2, MRCORD(23) ) ,
```

FILMED FROM BEST AVAILABLE COPY


```
     4     (MEDIA , MRCORD(24) ) , (TYPE , MRCORD(25) )  ,
     5     (QUEST , MRCORD(26)  ) , (PRES(1) , MRCORD(27) ) ,
     6     (COURSE, MRCORD(40) )  , (QTO  , QSTION(6) )  ,
     7     ( GMOD , QSTION(7)  )  , (QPOINT , QSTION(8) )
           INTEGER LIST(2)
           INTEGER DATE(6)
           INTEGER * 2 IQNUM
C BEGIN BY BUILDING A TABLE FOR DIRECT ACCESS TO THE QUESTION FILE.
           IREX = 0
           READ(ISYS'2) LPERM,NPERM
           IF(LPERM.GT.0) GO TO 200
           CALL ERROR(RNAME,0,1)
           RETURN
 200       CONTINUE
           MINLES = LPERM + 1
           READ(ISYS'3) PTRS
           READ(ISYS'4) NOS
C NOW CREATE THE TABLE
           DO 500 LES = 1,LPERM
           I1 = PTRS(LES)
           I2 = PTRS(LES) + NOS(LES) - 1
           DECK = 0
           DO 400 RNO = I1,I2
C READ THE HEADER
           READ(IHEAD'RNO) HRCORD
           DECK = DECK + 1
C PLACE THE INFO IN THE TABLE
           LTABLE(1,DECK,LES) = HRCORD(3)
           LTABLE(2,DECK,LES) = HRCORD(4)
           LTABLE(3,DECK,LES) = HRCORD(8)
           LTABLE(4,DECK,LES) = HRCORD(13)
 400       CONTINUE
 500       CONTINUE
C HAVING BUILT THE TABLE WE CAN BEGIN TO READ MBO RECORDS
 800       CONTINUE
           READ(INUNIT,801,END=1600) MRCORD
 801       FORMAT( I2 , I1 , I3 , I2 , 17A2 , I1, I1 , I2 , I2 , I2 , 1X ,
     2 12A2,A1, 2X, I2 )
           IREX = IREX + 1
C HAVING READ A RECORD CHECK ITS LESSON
           IF(LESSON.GE.1.AND.LESSON.LE.LPERM) GO TO 900
           I = LESSON
           J = SEGMNT
           K = TYPE
           CALL ERROR(RNAME,0,2,IREX,I,J,K)
           GO TO 800
 900       CONTINUE
C NOW SEARCH THE TABLE
           I1 = NOS(LESSON)
           DO 1500 I = 1,I1
           IF(LTABLE(1,I,LESSON).NE.SEGMNT) GO TO 1500
           IF(LTABLE(2,I,LESSON).NE.TYPE ) GO TO 1500
C HERE WE'VE FOUND THE TEST SO CHECK THE QUESTION NUMBER
           IF(QUEST.GE.1.AND.QUEST.LE.LTABLE(4,I,LESSON))GO TO 1000
C  THE QUESTION NUMBER IS INCORRECT
           J=LTABLE(4,I,LESSON)
           K=QUEST
           CALL ERROR(RNAME,0,3,IREX,J,K)
           GO TO 800
```

FILMED FROM BEST AVAILABLE COPY


```
C CHECK THE MBO AND TO NUMBERS
 1000     IF(TO.GT.0.AND.TO.LE.200.AND.MBO.GE.0) GO TO 1100
          J = TO
          K = MBO
          CALL ERROR(RNAME,0,4,IREX,J,K)
          GO TO 800
C ALL O.K. SO PLACE INFO ON DISK
 1100     RNO = LTABLE(3,I,LESSON) + QUEST - 1
          IF(LESSON.LT.MINLES) MINLES=LESSON
          READ(IQUEST'RNO) QSTION
          QTO = TO
          QMBO = MBO
          QSTION(13) = SKIL1
          QSTION(14) = SKIL2
          QSTION(15) = MEDIA
          DO 1200 J = 1,17
          QSTION(J + 40) = KEY(J)
 1200     CONTINUE
          DO 1300 J = 1,13
          QSTION(J + 57 ) = PRES(J)
 1300     CONTINUE
          WRITE(IQUEST'RNO) QSTION
          GO TO 800
 1500     CONTINUE
C IF WE FALL THROUGH HERE NO MATCH WAS FOUND
          I = LESSON
          J = SEGMNT
          K = TYPE
          CALL ERROR(RNAME,0,5,IREX,I,J,K)
          GO TO 800
C
C THIS WHOLE SCHMAGEGY IS TO DETERMINE WHICH LESSON TO BEGIN CHAINING.
C IT WILL BE EITHER THE FIRST UNCHAINED LESSON OR THE LOWEST LESSON FOR
C WHICH AN MBO WAS READ,. WHICHEVER IS LOWEST
C
C FIRST DETERMINE LAST UNCHAINED LESSON, AND LAST USED DIRECTORY RECORD
 1600     CONTINUE
          IDREC=1
          LESS = 1
          IF(LEVEL.NE.0) GO TO 2000
          DO 1700 I = 1 , LPERM
          J = PTRS(I)
          READ(IHEAD'J) HRCORD
          IF( HRCORD(7).EQ.0) GO TO 1800
          LESS = I
          IDREC = HRCORD(7)
 1700     CONTINUE
 1800     CONTINUE
C IF THIS THE LOWEST LESSON FOR WHICH MBO CARDS WERE READ WE HAVE
C THE CORRECT PARAMETERS SO WE CAN GO CHAIN
          IF(LESS.LE.MINLES) GO TO 2000
C WE'VE REPLACED OR ADDED MBO INFORMATION TO THE ALREADY CHAINED PORTION
C OF THE FILES SO WE MUST RECHAIN FROM MINLES
          LESS = MINLES
          II = PTRS(LESS)
          READ(IHEAD'II) HRCORD
          IDREC = HRCORD(7)
C IF NO RECORDS WERE READ AND ALL WERE CHAINED , RETURN
 2000     CONTINUE
```

FILMED FROM BEST AVAILABLE COPY


```
      IF(LESS.GT.LPERM) GO TO 7000
C
C THE NEXT AVAILABLE DIRECTORY RECORD IS IN IDREC
C THE LESSON TO CHAIN IS IN LESS
C
C CLEAR THE DIRECTORY RECORD
      DO 2100 I = 1,200
 2100 DRCTRY(I) = 0
C
C DETERMINE THE 1ST AND LAST RECORDS OF THE QUESTION FILE TO BE CHAINED
      I1 = PTRS(LESS)
      I2 = PTRS(LPERM) + NOS(LPERM) - 1
      READ(IHEAD'I1) HRCORD
      I1 = HRCORD(8)
      READ(IHEAD'I2) HRCORD
      I2 = HRCORD(8) + HRCORD(13) -1
C NOW PLACE MBO INFORMATION IN THE CHAINING AREA
C RECORD NUMBER IS POSITION IN THE FILE
      DO 2200 I = I1 , I2
      READ(IQUEST'I) QSTION
      CHAIN(1,I) = QTO
      CHAIN(2,I) = QMBO
      CHAIN(3,I) = 0
 2200 CONTINUE
C WE'RE ALL SET SO START TO CHAIN
C NOW WE START CHAINING
C
C DO LOOP FOR ALL LESSONS
 2400 CONTINUE
      DO 5500 LS = LESS,LPERM
      IFRST = PTRS(LS)
      ILAST = PTRS(LS) + NOS(LS) - 1
C DO LOOP FOR EACH HEADER
      DO 5000 HEAD = IFRST , ILAST
      READ(IHEAD'HEAD) HRCORD
C NOW WE BEGIN TO RUN THROUGH CHAIN FOR THE RECORDS SPECIFIED BY THE
C HEADER, FIRST SET UP THE DO LOOP
      IR1 = HRCORD(8)
      IR2 = HRCORD(13) + IR1 -1
      DO 4500 RNO = IR1,IR2
C IF NO TO NUMBER IGNORE RECORD
      TO = CHAIN(1,RNO)
      IF(TO.EQ.0)GO TO 4500
C BUT IF TO, THEN SEE IF FIRST OF THAT NUMBER
      MBO = CHAIN(2,RNO)
      IF(DRCTRY(TO).NE.0) GO TO 2800
C ITS THE FIRST IN THE CHAIN, SO PUT HEAD POINTER TO POINT TO IT
      DRCTRY(TO) = RNO
      GO TO 4500
 2800 CONTINUE
C ITS NOT THE FIRST, SO WE MUST SORT TO SET UP CHAIN IN MBO ORDER
C FIRST SEE IF THE NEW MBO NO IS LESS THAN THAT OF THE TOP OF THE CHAIN
      I  = DRCTRY(TO)
      IF(MBO.GE.CHAIN(2,I)) GO TO 3000
C IT IS SO PLACE,NEW RECORD AT THE TOP OF THE CHAIN
      CHAIN(3,RNO) = I
      DRCTRY(TO) = RNO
      GO TO 4500
 3000 CONTINUE
```

FILMED FROM BEST AVAILABLE COPY


```
C IT DOESN'T BELONG AT THE TOP OF THE STACK SO PLACE IT SOMEWHERE ELSE
C INITIALIZE FOR THE LOOP
      POINT1 = DRCTRY(10)
 3400 POINT2 = CHAIN(3,POINT1)
C SEE IF THE END OF THE CHAIN
      IF(POINT2.NE.0) GO TO 3800
C YES AT END OF CHAIN SO PLACE THERE
      CHAIN(3,POINT1) = RND
      GO TO 4500
C NOT AT END SO SEE IF IT BELONGS HERE
 3800 IF(MBO  .GE. CHAIN(2,POINT2) ) GO TO 4200
C YES IT BELONGS BETWEEN POINT1 AND POINT2
      CHAIN(3,RND) = POINT2
      CHAIN(3,POINT1) = RND
      GO TO 4500
 4200 CONTINUE
C NOT YET SO SEARCH FURTHER
      POINT1 = POINT2
      GO TO 3400
C THIS CLOSES THE CHAINING DO LOOP
 4500 CONTINUE
C NOW REWRITE THE HEADER WITH ITS DIRECTORY POINTER
      HRCORD(7) = IDREC
      WRITE(1HEAD'HEAD) HRCORD
C CLOSE THE LESSON DO LOOP
 5000 CONTINUE
C NOW START A NEW DIRECTORY RECORD, BUT IF NAVY ONLY AT THE END OF THE
C COURSE.
      IF (LEVEL.EQ.0) GO TO 5300
      IF(LS.NE.LPERM) GO TO 5500
 5300 CONTINUE
      WRITE(1DIR'IDREC) DRCTRY
      IDREC = IDREC + 1
      DO 5200 I = 1,200
 5200 DRCTRY(I) = 0
C WE'RE DONE , CLOSE MAJOR LOOP
 5500 CONTINUE
C FINISHED WITH THE CHAINING
C NOW MAKE A PASS THROUGH THE QUESTION FILE TO PLACE POINTERS
C IN THE RECORDS
      I1 = PTRS(LESS)
      I2 = PTRS(LPERM) + NOS(LPERM) - 1
      READ(1HEAD'I1) HRCORD
      I1 = HRCORD(8)
      READ(1HEAD'I2) HRCORD
      I2 = HRCORD(8) + HRCORD(13) - 1
C NOW PLACE POINTERS IN THE QUESTION FILE
      DO 5700 RND = I1 , I2
      READ(1QUEST'RND) QSTION
      QPOINT = CHAIN(3,RND)
      WRITE(1QUEST'RND) QSTION
 5700 CONTINUE
C  REWRITE NO. OF DIRECTORY RECORDS
      IDREC = IDREC - 1
      WRITE(1SYS'6) IDREC
C LIST THE FILE IN MBO ORDER
      ENTRY MBOLST
 7000 CONTINUE
      CALL INFO(DATE)
```

FILMED FROM BEST AVAILABLE COPY


```
      READ(ISYS'6) IDREC
      IPAGE = 0
      IREA = 0
      IF(IDREC.LE.0) GO TO 8600
C LOOP FOR EACH DIRECTORY RECORD
      DO 8500 IDR = 1, IDREC
      READ(IDIR'IDR) DRCTRY
      IN = 0
C LOOP FOR EACH TU
      DO 8000 TO = 1,55
      RNO = DRCTRY(TO)
 7300 IF(RNO.LE.0) GO TO 8000
      IF(IN.GT.0) GO TO 7500
      IPAGE = IPAGE + 1
      WRITE(IPRINT,7001) DATE,IDR,IPAGE
      WRITE(IPRINT,9302)
      IN = 10
 7500 CONTINUE
      IREA = IREA + 1
      READ(IQUEST'RNO) QSTION
      CALL LSTCDE(QSTION(12),LIST)
      WRITE(IPRINT,9001) (QSTION(L),L=1,3) , (QSTION(L),L=5,7)  ,
     2 LIST ,  (QSTION(L),L=13,15) , (QSTION(L),L=21,70)
      RNO= QPOINT
      IN = IN - 1
      GO TO 7300
 8000 CONTINUE
 8500 CONTINUE
 8600 CONTINUE
      WRITE(IPRINT,9701) IREA
      RETURN
C LIST THE FILE IN QUESTION ORDER
      ENTRY QLIST
 9000 CALL INFO(DATE)
C FIND NO. OF QUESTION RECORDS
      READ(ISYS'5) IQNUM
      IF(IQNUM.LE.0) GO TO 9700
C READ AND WRITE EACH RECORD
      IN = 0
      IPAGE = 0
      DO 9500 RNO = 1, IQNUM
      IF(IN.GT.0) GO TO 9300
      IPAGE = IPAGE + 1
      WRITE(IPRINT,9301) DATE, IPAGE
      WRITE(IPRINT,9302)
      IN = 10
 9300 CONTINUE
      READ(IQUEST'RNO) QSTION
      CALL LSTCDE(QSTION(12),LIST)
      WRITE(IPRINT,9001) (QSTION(L),L=1,3) , (QSTION(L),L=5,7)  ,
     2 LIST ,  (QSTION(L),L=13,15) , (QSTION(L),L=21,70)
      IN = IN - 1
 9500 CONTINUE
 9700 CONTINUE
      WRITE(IPRINT,9701) IQNUM
      RETURN
 7001 FORMAT(1H1 ,  7X , '***  ','AIMS MOD LISTING',
     2       '  ***',  56X    ,'JOB', 2X , 2A4 ,   2A ,
     3  2A4  ,  1X,  2A4  /    13X  ,'SCOPE NUMBER',  I4  ,
```

FILMED FROM BEST AVAILABLE COPY

```
        AIMS ### SCORE STATEMENT LISTING

     4          02/   .*PAGE*  .  14 ////
 9001    FORMAT##2X , I2 , 3A , I2 , 2A , I2 , 2A ,I2 , 3X , I2 , 2A  ,  *
     2  I2, 2A , 2A9 , 2A , I3 ,  3X ,  I2 ,  3X , I2 ,  2A , I3 ,
     3  2X , I3 , 2X  , I3 ,  2X  ,   I3 , 3X , I3  ,3X  , I3  , 3A,
     4  I3 , 2X , I3 , 1X ,  I3  ,  I2EI3, I3 )  /END ,I#A2, 4A  ,
     5  I2A2,A1   / )
 9302    FORMAT( 1X , *I###* , 1X , *SCO* ,  1X , *SP* , 1X , *GRI* ,
     2  1X , *IO* , 1X , *EO* , 1X , *CORRECT* ,  1X , *SKL* , 1X ,
     3  *SKL* ,  1X , *MUSA* , 1X , *PCI* ,   1X ,  *WALK* , 1X , *SEQ* ,
     4  1X  ,  *RRC* , 3A  , *AVG* , 3X  , *#* , 3A , *R* , 3X  , *ME* ,
     5  2X , *ECHO* , 1X , *RESPONSE COUNTERS*   /
     6  23A , *ANSWER* , 5X , *I* ,  3A , *2* , 2A , *SKILL* ,  4A  ,
     /  *DEV* , 1A , *AI* ,  1A , *GP* , 1X  , *AG* , 1A , *GP* , 1A ,
     8  *AI* ,   1A , *GP*  , 1X , *AG* , 1A , *GP* ,  1A , *GP* ,
     9  1X ,  *GRP* ,  1X ,  *SEA* , 1X , *A* , 2X ,  *B* ,  2X , *C* ,
     A  3X , *D* , 3X ,  *E* , 3X , *F* , 3X , *G* , 3X ,  *H* ,  3X ,
     B  *I* , 3X  ,  *J*  /  /   )
 9301    FORMAT(1H1 , 2X , *### *, *ABMS QUESTION LISTING* ,
     2       *  ###*,  3I 4  , *JOB*, 2X,  2A4 , 2A  ,
     3  2A4 ,  1X ,  2A4   /   9X ,    *PAGE* , I4  ////
 9701    FORMAT(1X,     3I3 ,  I3X        ,
     2  *TAKE ARE* , I4 ,  * RECORDS IN THIS FILE* )
         END
//*
 PHASE AIRSPACE,AIRSHEAD
// EXEC FFORTRAN
         SUBROUTINE PAGE(SECTION)
         COMMON/#ALES/IN(13)
         COMMON/STATE#/ILE(3)
         INTEGER DECKS
         INTEGER * 2 ANSWER(10,10),SECTION(10),SS(10,10),NO,IO
     2   SAMPLE(60),SDE#(60),MEAN(60),MIN(60),MAX(60),ICPLY(3),IO,
     3   HEADER(5,20)
         EQUIVALENCE(RLPLY(1),I3,SIGNAL(6,1))
     2   DIMENSION(1,1),SECTION(10,60) )
         CALL OPSYS('LOAD','AIRSPAC2')
         CALL OPSYS('LOAD','AIRSTAPE')
         CALL PROCESS(SESSION,DECKS,ANSWER,QSTION,REPLY,HEADER,
     2 SAMPLE,SDEV,MEAN,MIN,MAX )
         CALL OPSYS('LOAD','AIRSPACE')
         CALL PROC2(SESSION,DECKS,ANSWER,SECOND,QSTION,
     2 SAMPLE,SDEV,MEAN,MIN,MAX )
         RETURN
         END
//*
 PHASE AIRSPAC2,*
// EXEC FFORTRAN
         SUBROUTINE PROC1(SESSION,DECKS, ANSWER , QSTION, REPLY,HEADER ,
     2   SAMPLE , SDEV, MEAN , MIN, MAX )
C
C     THIS, THE FIRST OF THE TWO AIMS PROCESS ROUTINES, DOES THE ACTUAL
C GRADING, USING BOTH THE STUDENT'S RESPONSES AND THE CORRECT ANSWERS.
C THIS ROUTINE WILL CALCULATE A GRADE FOR EACH DECK IN THE LESSON, AS
C WELL AS THE MINIMUM, PARAMETER, MEAN, SED, DEV, OF EACH OF THE GRADES.
C     IN ADDITION THE VALIDITY, AND RESPONSE COUNTS FOR THE QUESTIONS
C ARE ALL DONE IN CORE IN THIS ROUTINE, AND PASSED TO THE 2ND PROCESS
C ROUTINE AS PARAMETERS. THIS ROUTINE WILL CREATE A SCRATCH FILE OF THE
C RESPONSE DATA, WHICH WILL BE USED IN THE 2ND OF THE PROCESS ROUTINES
C TO GET COUNTS, FOR EACH QUESTION, OF THE STUDENTS ABOVE THE MEAN AND OF
```

FILMED FROM BEST AVAILABLE COPY

```
      ARPS III SOURCE STATEMENT LISTING

C IF PLOT, AND IMUSE BELOW WHO GOT IT RIGHT.
C     THE FIRST SECTION WILL SET UP FOR I/O, THAT IS FIND THE CORRECT
C LESSON ON THE TAPE, AND READ THE APPROPRIATE HEADERS.
      COMMON/SYSTEM/RLESS,NDECK,NREA,NOUEST,NSTUD
      COMMON/FILES/IW(2),IPRINT,IRESP,ISCRCH, IHEAD,IH2(2),
     2    ISTART, ISCORE ,  IX3 ,  ISYS , IH9(3)
      INTEGER *2 LPCNP ,NUSE(50),  PTAS(50) , HEADER(65,10) ,MASTER,
     2  NOPCH,COUNT(111), SCRTCH(65)
      INTEGER PHAPLE(2),FLAG  , DECKS , TEST  , TYPE , STUENT, NOUST,
     2   SELECT, RSELCT , RATE , AMONG , AMONG2 , GST
      DATA PHAPL/'PRUE','ESS'/
      INTEGER * 2 REPLY(60),IQ1 , QSTION(10,40,10) , ANSWER(10,150)  ,
     2  MEAN(60) , SDEV(60) ,SAPPLL(60),PAX(60) , RIM(60), ANS, NSP
      REAL * 8       A , B , C , SUMSQ(10) , SUM(10)
      INTEGER ENDFIL , COUNTR, LNI , PLINER , GRD
      INTEGER * 2 MISSED
      DATA MISSED/-.09/
      ENDUNG(2) = E * 0.5
C
C SEE IF THIS LESSON IS IN PERMANENT
C
      READ(ISYS'2) LPERM
      IF(LESSON.GT.0.AND.LESSON.LE.LPERM) GO TO 200
C IF ILEGAL, SO CAN'T BE GRADED
      I = LESSON
      J = LPERM
      CALL ERROR(FRAME,0,I,I,J)
      RETURN
C
C THE NUMBER IS LEGAL SO SEARCH THE TAPE FOR THE CORRECT RECORD
  200 CONTINUE
      CALL TAPSRC(LESSON,FLAG)
C  IF NOT FOUND ALREADY AN ERROR MESSAGE SO JUST RETURN
      IF(FLAG.NE.0) RETURN
C
C NOW READ THE MAP OF THE HEADER FILE , AND THEN THE HEADERS
C
      READ(ISYS'3) PERS
      READ(ISYS'4) NUS
      DECKS = NUSE(LESSON)
      II = PERSE(LESSON)
      I2 = II + NUSE(LESSON) - 1
      J = 0
      DO 400 I = II , I2
      J = J + 1
      READ(IHEAD'I) (HEADER(K,J),K =1,65)
  400 CONTINUE
C
C WE WILL NOW INITIALIZE VARIABLES FOR ACCUMULATION, ETC.
C
C INITIALIZE QUESTION COUNTERS
      DO  600 K = 1, 10
      DO  600 J = 1, 40
      DO  600 I = 1,87
      QSTION(I,J,K) = 0
  600 CONTINUE
      DO 700 I = 1 , 10
      SUMSQ(I) = 0.0
      SUM(I) = 0.0
PAGE 40
```

FILMED FROM BEST AVAILABLE COPY


```
  700     CONTINUE
          DO  800 I = 1,60
          MAX(I) =  -1000
          MIN(I) =  10000
          SDEV(I)= 0
          SAMPLE(I) = 0
          MEAN(I) = 0
  800     CONTINUE
          ENDFIL = 0
C
C READ THE NUMBER OF STUDENTS
C
          READ(ISYS'1) ROSTER
          IF(ROSTER.GT.0) GO TO 850
          CALL ERROR(PNAME,0,0)
          RETURN
  850     CONTINUE
C THIS DO LOOP IS FOR ALL STUDENTS
C
          DO 6500 STDENT = 1, ROSTER
          DO 875 I = 1,10
          ANSWER(I,STDENT) = 0
  875     CONTINUE
          COUNTR= 0
          LRI = 0
          POINTR = 1
          DO 6250 TEST = 1 , DECKS
C GET SOME DATA IF ANY
C IF POINTR  IS LE COUNTR WE HAVE DATA
  900     IF( POINTR. LE . COUNTR) GO TO 1500
C IF NOT GET DATA IF POSSIBLE
C FIRST CHECK FOR EOF, 6000 IS MISSING DATA
          IF(ENDFIL . NE. 0) GO TO 6000
C CHECK THE LAST RECORD INDICATOR (LRI) TO SEE IF THERE ARE MORE
C RECORDS FOR THIS STUDENT
          IF( LRI . NE. 0 ) GO TO 6000
          CALL IPDATA( STDENT , FLAG , LRI ,COUNTR, REPLY)
C IF FLAG = 0 THE DATA WAS OBTAINED
          IF(FLAG . NE . 0 ) GO TO 1000
          POINTR = 1
          GO TO  1500
 1000     CONTINUE
C NO DATA WAS OBTAINED
C FLAG = 1 , NO DATA
C FLAG = 2 , END OF FILE
C FLAG = 3 , SORT ERROR, IPDATA GAVE ERROR MESSAGE , SO JUST RETURN
          GO TO(6000 ,1400 , 1200) , FLAG
 1200     CONTINUE
          RETURN
 1400     CONTINUE
          ENDFIL = 1
          GO TO 6000
 1500     CONTINUE
C WE HAVE SOME DATA, BUT IS IT THE RIGHT DECK
          IF( REPLY(3,POINTR) - HEADER(3,TEST) ) 1700 , 1600, 6000
C RIGHT SEGMENT , WHAT ABOUT TYPE
 1600     CONTINUE
          IF( REPLY(4,POINTR) - HEADER(4,TEST) ) 1700, 1600, 6000
C ILLEGAL DATA RECORD
```

FILMED FROM BEST AVAILABLE COPY


```
 1700     CONTINUE
          I = REPLY(1,POINTR)
          J = REPLY(2,POINTR)
          K = REPLY(3,POINTR)
          L = REPLY(4,POINTR)
          CALL ERROR(PNAME,0,5,I,J,K,L)
          POINTR = POINTR + 1
          GO TO 900
C
C WE'VE GOT THE CORRECT DATA RECORD, SO CALCULATE A GRADE
 1800     CONTINUE
C WE NOW ZAP THROUGH A TEST AND GATHER BASIC INFORMATION
          TYPE = HEADER(4,TEST)
          NOQST = HEADER(13,TEST)
          SELECT = 0
          RSELCT = 0
          RITE   = 0
          NOGRD = 0
          DO 2000  I   = 1 , 17
          SCRTCH(I) = REPLY(I,POINTR)
 2000     CONTINUE
          DO 3000 QST = 1, NOQST
          QNO = QST + 17
          ANS = HEADER(QNO,TEST)
          RSP = REPLY(QNO,POINTR)
          NO = IPASS(ANS,RSP,NOPCH,COUNT)
C COUNT THE RESPONSES
          DO 2100 I = 1, 11
          QSTION(I,QST,TEST) = QSTION(I,QST,TEST) + COUNT(I)
 2100     CONTINUE
C IF ANSWER IS BLANK'DON'T GRADE THE QUESTION
          IF( ANS .EQ . 1)  GO TO 2600
C SEE IF HE GOT A  RIGHT RESPONSE
          IF( NO. EQ .0 ) GO TO 2600
          RSELCT = RSELCT + 1
          IF(NO. EQ. NOPCH) RITE = RITE + 1
 2500     CONTINUE
          SELECT = SELECT + ( NOPCH - NO)
          IF(RSP.NE.1) NOGRD = NOGRD + 1
 2600     CONTINUE
          SCRTCH(QNO) = NO
 3000     CONTINUE
C
C WE NOW HAVE A GRADE , SO PERFORM FURTHER CALCULATIONS
C
C  HERE WE HAVE ALL THE INFORMATION NECESSARY TO CALCULATE
C  A GRADE AND SO WE SHALL DO SO
C  EACH STATEMENT IS FOR A DIFFT ALGORITHM AND THERE ALSO SHALL
C  BE DIFFERENT ALGORITHMS FOR NAVY AND HEW
       GO TO(3200,3400,3600,3800,3400,4200,4400,4600,4800),TYPE
 3400     CONTINUE
          C = RITE
          Q = HEADER(12,TEST)
          GRADE =  (C / Q )  * 100.0
          GO TO 5000
 3600     CONTINUE
          GRADE = 0.0
          DO 3700 QST = 1 , NOQST
          QNO = QST + 17
```

FILMED FROM BEST AVAILABLE COPY


```
            RSP = REPLY(QNO,POINTR)
            IF(RSP.NE.3) GO TO 3640
            GRADE = GRADE +  10.0
            GO TO 3700
 3640       IF(RSP.NE.5) GO TO 3660
            GRADE = GRADE + 8.0
            GO TO 3700
 3660       IF(RSP.NE.9) GO TO 3680
            GRADE = GRADE + 6.0
            GO TO 3700
 3680       IF(RSP.EQ.17) GRADE = GRADE + 3.0
 3700       CONTINUE
            GRADE = GRADE / ( HEADER(12, TEST ) ) * 10.0
            GO TO 5000
 3800       CONTINUE
            C = XSELCT
            Q = HEADER(12,TEST)
            H = SELECT
            P = HEADER(11,TEST) - Q
            X = Q * ( H / P )
            GRADE = ( C - X ) / Q * 100.0
            GO TO 5000
 3200       CONTINUE
 4000       CONTINUE
 4200       CONTINUE
 4400       CONTINUE
 4600       CONTINUE
 4800       CONTINUE
            GRADE = 0
 5000       CONTINUE
            INTR =  IROUND(GRADE)
            IF(INTR .GT. MAX(TEST) ) MAX(TEST) = INTR
            IF(INTR .LT. MIN(TEST) ) MIN(TEST) = INTR
            SAMPLE(TEST) = SAMPLE(TEST) + 1
            SUM( TEST ) = SUM( TEST) + GRADE
            SUMSQ( TEST) = SUMSQ( TEST) + ( GRADE * GRADE )
            ANSWER(TEST,STDENT) = INTR
            SCRTCH(11) = INTR
            IREC = (( STDENT - 1) * DECKS)  + TEST
            WRITE(ISCRCH'IREC) SCRTCH
C NOWGET COUNTS AND AVGS FOR QSTION VALIDITY
            DO 5500 QST = 1, NOQST
            QNO = QST + 17
            IF(SCRTCH(QNO).LE.0)GO TO 5200
            QSTION(12,QST,TEST) = QSTION(12,QST,TEST) + INTR
            QSTION(14,QST,TEST) = QSTION(14,QST,TEST) + 1
            GO TO 5500
 5200       CONTINUE
C  NO HE'S IN THE WRONG GROUP
            QSTION(13,QST,TEST) = QSTION(13,QST,TEST) + INTR
            QSTION(15,QST,TEST) = QSTION(15,QST,TEST) + 1
 5500       CONTINUE
            POINTR = POINTR + 1
            IF(TEST.NE.DECKS) GO TO 6250
            IF(POINTR.GT.COUNTR) GO TO 6250
            DO 5750 M = POINTR,COUNTR
            I = REPLY(1,M)
            J = REPLY(2,M)
            K = REPLY(3,M)
```

FILMED FROM BEST AVAILABLE COPY


```
            L = REPLY(4,M)
            CALL ERROR(PNAME,0,5,I,J,K,L)
 5750       CONTINUE
            GO TO 6250
C MISSING DATA
 6000       CONTINUE
            ANSWER(TEST,STUDNT) = -999
            IREC = (( STUDNT - 1) * DECKS)  + TEST
            WRITE(ISCRCH'IREC) MISSED
 6250       CONTINUE
 6500       CONTINUE
            DO 8000 TEST = 1, DECKS
            N = SAMPLE(TEST)
            IF(N.LE.0) GO TO 7500
            A =  SUM(TEST)
            G =  A / N
            MEAN(TEST) = IROUND(G)
            IF(N.GT.1) GO TO 7000
            SDEV(TEST) = 0
            GO TO 8000
 7000       CONTINUE
C CALCULATE STANDARD DEVIATIONS
            B = A * A
            C = N * SUMSQ(TEST)
            S = N * (N- 1)
           SS = ( C - B)  / S
           SD = SQRT(SS)
           SDEV(TEST) = IROUND(SD)
           GO TO 8000
 7500      CONTINUE
           MEAN(TEST) = -999
           SDEV(TEST) = -999
           MAX(TEST)  = -999
           MIN(TEST)  = -999
 8000      CONTINUE
           RETURN
           END
/*
// ASSGN SYSSLB,X'191'
// EXEC ASSEMBLY
IPASS     START
          COME  ENTER
CORECT    EQU   4
STUDNT    EQU   6
ENTER     L     3,0(0,1)          1ST PAR IS CORECT ANS
          LH    5,0(0,3)          PLACE CORECT ANS IN GPR5
          L     3,4(0,1)          2ND PAR IS STUDENT'S RESPONSE
          LH    7,0(0,3)          PLACE STUDENT ANS IN GPR7
          NR    5,7               CHECK CORRECT BITS
          L     11,12(0,1)        LOAD GPR11 WITH ADDR OF COUNTS
*                                 COUNTS ARE 4TH PARAMETER
          USING CNTAR,11          TELL ASSEMBLER
          MVI   BLANK,X'00'       CLEAR COUNT AREA
          MVC   BLANK+1(21),BLANK
          SR    0,0               CLEAR GPR0 FOR TOTAL CORRECT
          SR    3,3               CLEAR GPR3 FOR TOTAL
          SLL   5,21              LEFT SHIFT TO SET UP FOR COUNTS
          SLL   7,21
*                                 THIS LOOP BEGINS THE COUNTING
```

FILMED FROM BEST AVAILABLE COPY


```
*                                                    SET UP FOR THE LOOP
         LA     10,16(0,0)                           THIS IS OUR INDEX GPR
         L      6,=F'-2'                             THIS IS THE INCREMENT
         L      9,=F'-1'                             THIS IS THE COMPARAND
LOOP     SR     CORECT,CORECT                        CLEAR THE RI ORDER GPR'S
         SR     STUDNT,STUDNT
         SLDL   CORECT,1                             PLACE A BIT IN THE GPR
         SLDL   STUDNT,1                             ALSO FOR HIS RESPONSE
         AR     0,CORECT                             COUNT THE CORRECT BITS
         AR     3,STUDNT                             COUNT THE STUDENT RESPONSE
         STH    STUDNT,COUNTS(10)                    AND COUNT THE RESPONSE
         BXH    10,6,LOOP                            FINISH LOOP
         C      3,=F'0'
         BC     X'2',SOME                            IF NONE OTHERS,THEN A BLANK
         L      4,=F'1'
         STH    4,BLANK                              COUNT THE BLANK
SOME     L      4,8(0,1)                             GET ADR OF 3RD PARAMETER
         STH    3,0(0,4)                             PLACE RESPONSE CNT IN 3RD PAR
         HOME
CNTAR    DSECT
BLANK    DS     1H
COUNTS   DS     10H
         END
/*
 PHASE AIMSTAPE,*
// EXEC FFORTRAN
         SUBROUTINE TAPSVC(LESSON,FLAG)
C
C THE MAIN ENTRY POINT TO THIS ROUTINE WILL SEARCH THE TAPE FOR
C THE REQUESTED LESSON, IF FOUND, FLAG WILL BE RETURNED AS 0
C OTHERWIDE A 1 WILL INDICATE THAT NO DATA WAS FOUND FOR THE LESSON, OR
C A 2 WILL INDICATE AN UNSORTED TAPE
         COMMON/FILES/IW(3),ITAPE,IW1(11)
         INTEGER ENDFIL, RDFLAG , RCOUNT, FLAG, COUNT, STUDNT
         INTEGER * 2  SORT(4), RSPONS(65) , REPLY(65,10)
         INTEGER PNAME(2)
         DATA PNAME/'PROC','ESS'/
C OPEN THE FILE AND INITIALIZE
         REWIND ITAPE
         ENDFIL = 0
         DO 100 I = 1,4
         SORT(I) = 0
  100    CONTINUE
C SET  THE RDFLAG TO OFF
         RDFLAG = 0
         RCOUNT = 0
         LESS = LESSON
         REWIND ITAPE
  500    READ(ITAPE,END=1000) RSPONS
         RCOUNT = RCOUNT + 1
C BEGIN TO CHECK FOR SORT ERROR
         DO 700 I = 1 , 4
         IF(SORT(I) - RSPONS(I) )  800 ,700, 600
  600    CONTINUE
C THIS IS A SORT ERROR
         J = SORT( I )
         K = RSPONS( I )
         CALL ERROR(PNAME,0,2,RCOUNT,I,J,K)
         FLAG = 2
```

FILMED FROM BEST AVAILABLE COPY


```
            RETURN
  700       CONTINUE
C HERE WE HAVE 2 IDENTICAL RECORDS, THIS IS A BADNESS
            I = RSPONS(2)
            J = RSPONS(3)
            K = RSPONS(4)
            CALL ERROR(PNAME,0,3,RCOUNT,I,J,K)
            FLAG = 2
            RETURN
  800       CONTINUE
C NO SORT ERROR SAVE SORT PARAMETERS
            DO 900  J = 1,4
            SORT(J) = RSPONS(J)
  900       CONTINUE
C NOW CHECK THE LESSON
            IF(RSPONS(1) - LESSON)  500,950,1000
C THIS IS THE RIGHT LESSON
  950       CONTINUE
C BUT  IF ITS A HEADER SKIP IT
            IF(RSPONS(2).EQ.0) GO TO 500
C WE'RE HERE SO CAN RETURN
            FLAG = 0
            RETURN
C THE LESSON ISN'T ON THIS TAPE
 1000       CONTINUE
            ENDFIL = 1
            REWIND ITAPE
            I= LESSON
            J = SORT(1)
            CALL ERROR(PNAME,0,4,I,J)
            FLAG = 1
            RETURN
C THIS ENTRY SEARCHES FOR A GIVEN STUDENT'S DATA
C IF NONE IS FOUND FLAG WILL RETURN A 1, ON END OF FILE ( THAT IS THE
C FIRST CALL WITH NO DATA RETURNED FLAG WILL RETURN A 2
C
            ENTRY TPDATA(STUDNT,FLAG,LCI,COUNT,REPLY)
C SET COUNT TO 0 , I. E. NO DATA
            COUNT = 0
            LCI = 0
            FLAG = 0
            IF(ENDFIL.NE.0) GO TO 9000
C WE CAN READ SHOULD WE
 2000       CONTINUE
            IF(ROFLAG.EQ.0) GO TO 3000
            READ(ITAPE,END=8000) RSPONS
            IF(RSPONS(1).NE.LESS) GO TO 8000
            RCOUNT = RCOUNT + 1
C WE'VE READ A RECORD CHECK THE SORT
            DO 2200 I = 1,4
            IF(SORT(I) - RSPONS(I) ) 2300 , 2200, 2100
 2100       CONTINUE
C SORT ERROR
            J = SORT( I)
            K = RSPONS(I)
            CALL ERROR(PNAME,0,2,RCOUNT,I,J,K)
            FLAG = 3
            RETURN
 2200       CONTINUE
```

FILMED FROM BEST AVAILABLE COPY


```
C IDENTICAL RECORDS
          I = RSPONS(2)
          J = RSPONS(3)
          K = RSPONS(4)
          CALL ERROR(PNAME,0,3,RCOUNT,I,J,K)
          FLAG = 3
          RETURN
C SORT IS O.K.
 2300     CONTINUE
          DO 2400 J = 1, 4
          SORT(J) = RSPONS(J)
 2400     CONTINUE
C HERE WE'VE GOT DATA, FROM READING, OR LAST TIME
 3000     CONTINUE
          RDFLAG = 1
          IF( RSPONS(2) - STUDNT) 3200, 4000, 3500
C WE MUST SKIP, BUT NOTE THE FACT
 3200     I= RSPONS(2)
          J = RSPONS(3)
          K = RSPONS(4)
          L= RSPONS(1)
          CALL ERROR(PNAME,0,5,L,I,J,K)
          GO TO 2000
C NO MORE DATA FOR STUDENT
 3500     CONTINUE
          RDFLAG = 0
          LCI = 1
C WAS THERE ANY DATA BEFORE
          IF(COUNT.NE.0) RETURN
C NO SO STUDENT HAD NO DATA
          FLAG = 1
          RETURN
C THIS DATA BELONGS TO THIS STUDENT
C BUT IS THIS THE 11TH RECORD
 4000     CONTINUE
          IF(COUNT.LT.10) GO TO 4500
C    YES
          LCI = 0
          RDFLAG = 0
          RETURN
C NO, SO PLACE DATA IN ANS
 4500     COUNT = COUNT + 1
          DO 5000 I = 1, 65
          REPLY(I,COUNT) = RSPONS(I)
 5000     CONTINUE
C AND GO READ AGAIN
          GO TO 2000
C THERE IS NO MORE DATA
 8000     CONTINUE
          ENDFIL = 1
          IF(COUNT.EQ.0) GO TO 9000
          LCI = 1
          REWIND ITAPE
          RETURN
C WE WILL NOT READ AGAIN
 9000     CONTINUE
          LCI = 1
          FLAG = 2
          REWIND ITAPE
```

FILMED FROM BEST AVAILABLE COPY


```
          RETURN
          END
/*
 PHASE AIMSPRC3,AIMSPRC2
// EXEC FFORTRAN
          SUBROUTINE PROC2(LESSON,DECKS, ANSWER , SECOND , QSTION ,
     2    SAMPLE , SDEV, MEAN , MIN, MAX )
          REAL * 8 SUM(20) , SUMSQ(20) , A , B , C
          INTEGER * 2  ANSWER(10,150) , SECOND(10,150) , QSTION(18,48,10)
     2  , SAMPLE(60) , SDEV(60) , MEAN(60) , MIN(60) , MAX(60)
          INTEGER * 2 RECORD(65) , TABLE(2,10) ,PTRS(50) , IDENT(5)
       COMMON/SYSTEM/NLESS,NDECK,NREX,NQUEST,NSTUD
          COMMON/FILES/ IW1(4), ISCRCH , IHEAD, IW2 , IQUEST ,
     2    ISTDNT , ISCORE , IW3, ISYS , IW4(3)
          INTEGER * 2  CAPIN , ROSTER
          EQUIVALENCE(CAPIN,RECORD(25))
          INTEGER DECKS , STDENT , TEST , CNC, QST ,AVG , AVG2
C FIRST BUILD A TABLE (FROM THE HEADER FILE)TO DESCRIBE THE
C QUESTION FILE
C
          DO 250 I = 1,20
          SUM(I) = 0.0
          SUMSQ(I) = 0.0
  250     CONTINUE
          READ(ISYS'3) PTRS
          J = PTRS(LESSON)
          DO 500 I = 1 , DECKS
          READ(IHEAD'J) RECORD
          TABLE(1,I) = RECORD(8)
          TABLE(2,I) = RECORD(13)
          J = J + 1
  500     CONTINUE
C FIND THE NUMBER OF STUDENTS
          READ(ISYS'1) ROSTER
C  TWO   PASSES WILL BE NECESSARY TO OBTAIN ALL GRADES AS THE LAST
C  USES   MEANS FROM THE PREVIOUS PASS
C
C IN PASS 1 WE WILL 1)  COLLECT INFORMATION ON THE CAPABILITY INDICES
C                       FROM THE DISK
C                   2)  PERFORM COUNTS FOR HI AND LO GROUPS FOR QUESTION
C                       COUNTERS ( ALSO FROM DISK )
C                   3)  CALCULATE THE COMBINED SECONDARY GRADES,
C                       PERFORMANCE INDEX AND NET ACHEIVEMENT INDEX
C PASS 1
          DO 3000 STDENT = 1 , ROSTER
C FIRST GET THE CAPABILITY INDEX
          DO 750 I = 1,10
          SECOND(I,STDENT) = 0
  750     CONTINUE
          READ(ISTDNT'STDENT) RECORD
          SECOND(1,STDENT) = CAPIN
          IF(CAPIN.EQ.-999) GO TO 1000
          SAMPLE(11) = SAMPLE(11) + 1
          SUM(11) = SUM(11) + CAPIN
          SUMSQ(11) = SUMSQ(11) + (CAPIN * CAPIN )
          IF(CAPIN.GT.MAX(11) ) MAX(11) = CAPIN
          IF(CAPIN.LT.MIN(11) ) MIN(11) = CAPIN
C NOW CALCULATE THE COMBINED GRADES
C FIRST IS NET ACHEIVEMENT INDEX
```

FILMED FROM BEST AVAILABLE COPY


```
 1000     CONTINUE
      IF(ANSWER(1,STDENT).NE.-999.AND.ANSWER(2,STDENT).NE.-999)GO TO1200
         SECOND(2,STDENT) = -999
         SECOND(3,STDENT) = -999
         GO TO 1500
 1200     A = ANSWER(1,STDENT)
         GRADE     =    (A + ANSWER(2,STDENT)) / 2.0
         INTER  = IROUND(GRADE)
         SECOND(2,STDENT) = INTER
         SAMPLE(12) = SAMPLE(12) + 1
         SUM(12) = SUM(12) + GRADE
         SUMSQ(12) = SUMSQ(12) + (GRADE * GRADE)
         IF(INTER.LT.MIN(12)) MIN(12) = INTER
         IF(INTER.GT.MAX(12)) MAX(12) = INTER
C NOW CALCULATE ABSOLUT ACHEIVEMENT DEVIATION
         IF(CAPIN.NE.-999) GO TO 1400
         SECOND(3,STDENT) = -999
         GO TO 1500
 1400     CONTINUE
         GRADE = GRADE - CAPIN
         INTER = IROUND(GRADE)
         SECOND(3,STDENT) = INTER
         SAMPLE(13) = SAMPLE(13) + 1
         SUM(13) = SUM(13) + GRADE
         SUMSQ(13) = SUMSQ(13) + (GRADE * GRADE)
         IF(INTER.LT.MIN(13) ) MIN(13) = INTER
         IF(INTER.GT.MAX(13) ) MAX(13) = INTER
 1500     CONTINUE
C NOW CALCULATE PERFORMANCE INDEX
      IF(ANSWER(3,STDENT).NE.-999.AND.ANSWER(4,STDENT).NE.-999)GOTO 1700
         SECOND(5,STDENT) = -999
         SECOND(6,STDENT) = -999
         GO TO 2000
 1700     CONTINUE
         A = ANSWER(3,STDENT)
         GRADE =((3.0 * A) + ANSWER(4,STDENT)) / 4.0
         INTER = IROUND(GRADE)
         SECOND(5,STDENT) = INTER
         SAMPLE(15) = SAMPLE(15) + 1
         SUM(15) = SUM(15) + GRADE
         SUMSQ(15) = SUMSQ(15) + (GRADE * GRADE)
         IF(INTER.LT.MIN(15) ) MIN(15) = INTER
         IF(INTER.GT.MAX(15) ) MAX(15) = INTER
C NOW CALCULATE ABSOLUTE PERFORMANCE DEVIATION
         IF(CAPIN.NE.-999) GO TO 1900
         SECOND(6,STDENT) = -999
         GO TO 2000
 1900     CONTINUE
         GRADE = GRADE - CAPIN
         INTER = IROUND(GRADE)
         SECOND(6,STDENT) = INTER
         SAMPLE(16) = SAMPLE(16) + 1
         SUM(16) = SUM(16) + GRADE
         SUMSQ(16) = SUMSQ(16) + (GRADE * GRADE)
         IF(INTER.LT.MIN(16) ) MIN(16) = INTER
         IF(INTER.GT.MAX(16) ) MAX(16) = INTER
 2000     CONTINUE
C NOW WE'LL ZAP THROUGH THE PRIMARY GRADES IF THE STUDENT'S SCORE
C IS ABOVE OR = TO THE MEAN , WE'LL COUNT HIM IN THE RIGHT OR WRONG
```

FILMED FROM BEST AVAILABLE COPY


```
C QUESTION COUNTERS
      DO 2600 TEST = 1 , DECKS
      IF(ANSWER(TEST,STDENT).LT.MEAN(TEST) ) GO TO 2600
      IREC = ((STDENT - 1) * DECKS) + TEST
      READ(1SCRCH'IREC) RECORD
      NUM = TABLE(2,TEST)
      DO 2500 QNO = 1,NUM
      QST = QNO + 17
      IF(RECORD(QST).LT.1) GO TO 2250
      QSTION(16,QNO,TEST) = QSTION(16,QNO,TEST) + 1
      GO TO 2500
 2250 CONTINUE
      QSTION(17,QNO,TEST) = QSTION(17,QNO,TEST) + 1
 2500 CONTINUE
 2600 CONTINUE
 3000 CONTINUE
C NOW PREPATORY TO PASS 2 WE'LL CALCULATE THE MEAN OF THE
C ABSOLUTE ACHEIVEMENT DEVIATION
      IF (SAMPLE(13).GT.0) GO TO 4000
C NO ONE HAS ABS ACHV DEV
      DO 3500 I = 1,ROSTER
      SECOND(4,I) = - 888
 3500 CONTINUE
C NOW GO ON TO REL PREF DEV
      GO TO 5100
 4000 CONTINUE
      GRADE = SUM(13) / SAMPLE(13)
      AVG   = IROUND(GRADE)
C NOW CALCULATE RELATIVE ACHIEVEMENT DEVIATIONS
      DO 5000 STDENT = 1,ROSTER
      IF(SECOND(3,STDENT).NE.-999) GO TO 4500
      SECOND(4,STDENT) = -999
      GO TO 5000
 4500 CONTINUE
      GRADE = SECOND(3,STDENT) - AVG
      INTER = IROUND(GRADE)
      SECOND(4,STDENT) = INTER
      SAMPLE(14) = SAMPLE(14) + 1
      SUM(14) = SUM(14) + GRADE
      SUMSQ(14) = SUMSQ(14) + (GRADE * GRADE)
      IF(INTER.LT.MIN(14) ) MIN(14) = INTER
      IF(INTER.GT.MAX(14) ) MAX(14) = INTER
 5000 CONTINUE
C NOW MEAN OF THE ABSOLUTE PERFORMANCE DEVIATION
 5100 CONTINUE
      IF(SAMPLE(16).GT.0) GO TO 6000
C NO ABS PERF DEVS
      DO 5500 I = 1,ROSTER
      SECOND(7,I) = -888
 5500 CONTINUE
      GO TO 7100
 6000 CONTINUE
      GRADE = SUM(16) / SAMPLE(16)
      AVG   = IROUND(GRADE)
C NOW CALCULATE RELATIVE PERFORMANCE DEVIATION
      DO 7000 STDENT = 1,ROSTER
      IF(SECOND(6,STDENT).NE.-999) GO TO 6500
      SECOND(7,STDENT) = -999
      GO TO 7000
```

FILMED FROM BEST AVAILABLE COPY


```
 6500     CONTINUE
          GRADE = SECOND(6,STUDENT) - AVG
          INTER = IROUND(GRADE)
          SECOND(7,STUDENT) = INTER
          SAMPLE(17) = SAMPLE(17) + 1
          SUM(17) = SUM(17) + GRADE
          SUMSQ(17) = SUMSQ(17)+(GRADE * GRADE )
          IF(INTER.LT.MIN(17) ) MIN(17) = INTER
          IF(INTER.GT.MAX(17) ) MAX(17) = INTER
 7000     CONTINUE
C NOW CALCULATE MEANS, STD DEVS, ETC.
 7100     CONTINUE
          DO 8000 TEST = 11,17
          N = SAMPLE(TEST)
          IF(N.LE.0) GO TO 7500
          A = SUM(TEST)
          G = A/N
          MEAN(TEST) = IROUND(G)
          IF(N.GT.1) GO TO 7300
          SDEV(TEST) = 0
          GO TO 8000
 7300     CONTINUE
          B = A * A
          C = N * SUMSQ(TEST)
          S = N * (N-1)
          SS = ( C -B ) / S
          SD = SQRT(SS)
          SDEV(TEST) = IROUND(SD)
          GO TO 8000
 7500     CONTINUE
          MEAN(TEST) = -999
          SDEV(TEST) = -999
          MAX(TEST) = -999
          MIN(TEST) = -999
 8000     CONTINUE
C WE WILL NOW PLACE THE QUESTION COUNTERS ON DISK
          DO 10000 TEST = 1, DECKS
          NUM = TABLE(2,TEST)
          INK = TABLE(1,TEST)
          DO 10000 QST = 1, NUM
          IREC = QST * INK - 1
C READ A QUESTION RECORD
          READ(1QUEST'IREC) RECORD
          DO 8500 I = 1,11
          J = I * 29
          RECORD(J) = QSTION(I,QST,TEST)
 8500     CONTINUE
          DO 8600 I = 24,25
          J = I - 12
          I = QSTION(J+2,QST,TEST)
          IF(I.GT.0.99) GO TO 8600
          RECORD(I) = -999
          GO TO 8800
 8600     CONTINUE
          GRADE = QSTION(J,QST,TEST) / I
          RECORD(I) = IROUND(GRADE)
 8800     CONTINUE
          DO 9000 I = 14,17
          J = I * 12
```

FILMED FROM BEST AVAILABLE COPY


```
            RECORD(J) = SESSION(I,QST,TEST)
 9000       CONTINUE
C PER CENT RIGHT
            ENTER = RECORD(26) + RECORD(27)
            IF(ENTER.GT.0) GO TO 9100
            RECORD(28) = -999
            GO TO 9200
 9100       CONTINUE
            S = RECORD(26)
            GRADE = ( 65 / ENTER) * 100.0 )
            RECORD(28) = ERCURD(GRADE)
 9200       CONTINUE
            ENTER = SUEV(TEST)
            IF(ENTER.GT.0) GO TO 9300
            RECORD(22) = -999
            GO TO 9500
C VALIDITY
 9300       CONTINUE
            IF(RECORD(24).EQ.-999.OR.RECORD(25).EQ.-999) GO TO 9400
            S = RECORD(24) - RECORD(25)
            T = RECORD(26) + RECORD(27)
            II = RECORD(26) / T
            I2 = 1.0 - II
            I = II * I2
            GRADE =  ((SUM(I) / ENTER) * 100.0) * S
            RECORD(22) = ERCURD(GRADE)
            GO TO 9500
 9400       CONTINUE
            RECORD(22) = -999
 9500       CONTINUE
            RECORD(23) = SUEV(TEST)
            WRITE(4,QUEST*IREC) RECORD
10000       CONTINUE
C GET NME COURSE
            READ(1,FORMAT*1) RECORD
            IDENT(5) = RECORD(5)
            IDENT(1) = LESSON
            IDENT(2) = - 4
            IDENT(3) =  0
            IDENT(4) = 0
C NOW PLACE GRADES ON DISK
            NUMB = NSTUD * 5
            IREC =  (( LESSON - 1) * NUMB ) + 1
C WRITE SAMPLE SIZE
            WRITE(1,SCORE*IREC) IDENT , SAMPLE
C WRITE MEAN
            IDENT(2)= -3
            IREC = IREC + 1
            WRITE(1,SCORE*IREC) IDENT , MEAN
C WRITE MIN
            IDENT(2) = -2
            IREC = IREC + 1
            WRITE(1,SCORE*IREC) IDENT , MIN
C WRITE MAX
            IDENT(2) = -1
            IREC = IREC + 1
            WRITE(1,SCORE*IREC) IDENT , MAX
C WRITE SUEV
            IDENT(2) = 0
```

FILMED FROM BEST AVAILABLE COPY

```
      AIMS III SOURCE STATEMENT LISTING

            IREC = IREC + 1
            WRITE(ISCORE'IREC) IDENT , SDEV
C ALSO WRITE THE INDIVIDUAL SCORES AND RETURN
            DO 12000 STDENT = 1,ROSTER
            IDENT(2) = STDENT
            IREC = IREC + 1
            WRITE(ISCORE'IREC) IDENT , (ANSWER(I,STDENT),I=1,10 ) ,
     2      (SCORAD(J,STDENT),J = 1,10)
12000       CONTINUE
            RETURN
            END
            FUNCTION IROUND(E)
            IF(E.EQ.0.0) GO TO 1000
            IROUND = E + (0.5 * ( E / ABS(E ) ) )
            RETURN
 1000       CONTINUE
            IROUND = 0
            RETURN
            END
/*
 PHASE AIMSLIST,AIMSHEAD
// EXEC FORTRAN
            SUBROUTINE ALIST(LESSON)
C THIS ROUTINE PROVIDES THE USER WITH THE FACILITY TO LIST AN AIMS DATA
C TAPE. IT USES TAPSVC TO PROVIDE THE TAPE HANDLING AND ESTCDC
C TO FORMAT THE ANSWERS IN A USER READABLE FORMAT
C
            COMMON/FILES/IN(2),IOUT,IRESP,IMT,IHEAD,IN2(2),ISTDNT,IN3(2),
     2      ISYS,IN4(3)
            INTEGER LNAME(2),DATE(6)
            INTEGER * 2 PLESS , NSTUD
            INTEGER * 2 DATA(65,10) , NOS(50), PTRS(50)
            DATA LNAME/'LIST',' '/
            INTEGER SCRDI(2,2) , SLINE(2,10)  ,CHECK , BLANK ,ANS(2,40,10)
            DATA SCRDI/'SING','LE','DOUB','LE'/, BLANK/' '/
            INTEGER FLAG ,STDENT , COUNT , SLASH , DOT
            DATA SLASH/'/'/ , DOT/'.'/
            INTEGER * 2  MASTE(5) , NAME(13)
C CHECK THE LESSON NUMBER
            READ(ISYS'2) PLESS
            IF(LESSON.GT.0.AND.LESSON.LE.PLESS) GO TO 250
            I = LESSON
            J = PLESS
            CALL ERROR(LNAME,0,1,I,J)
            RETURN
  250       CONTINUE
            READ(ISYS'3) PTRS
            READ(ISYS'4) NOS
            I1 = PTRS(LESSON)
            I2 = NOS(LESSON) + I1 - 1
            NOHEAD = NOS(LESSON)
            J = 0
            DO 500 I = I1,I2
            J = J + 1
            READ(IHEAD'I) (DATA(K,J),K=1,65)
  500       CONTINUE
C WE'VE READ THE HEADERS NOW PRINT THEM
            IPAGE = 1
            CALL INFO(DATE)
PAGE 02
```

FILMED FROM BEST AVAILABLE COPY

```
      AIMS III SOURCE STATEMENT LISTING

            WRITE(IOUT,501) DATES,IPAGE
  501       FORMAT(1H1,6X,'***  AIMS LESSON LISTING  ***',54X,'JOB  ',2A4,
     2      2A, 2A4 ,  1X , 2A4 / 90X ,'PAGE',I4 )
C  WRITE OUT THE WHOLE THING
            WRITE(IOUT,99)  ( I, I = 1, NUMHEAD)
            WRITE(IOUT,88)  (DATA(1,I) , I = 1, NUMHEAD )
            WRITE(IOUT,89)
            WRITE(IOUT,91)  (DATA(3,I) , I = 1, NUMHEAD )
            WRITE(IOUT,92)  (DATA(4,I) , I = 1, NUMHEAD )
            WRITE(IOUT,93)  (DATA(5,I) , I = 1, NUMHEAD )
            WRITE(IOUT,96)  (DATA(6,I) , I = 1, NUMHEAD )
            DO 600 I = 1 , NUMHEAD
            K = DATA(10,I) + 1
            DO 600 J = 1 , 2
            SLINE(J,I)  = SORU(J,K)
  600       CONTINUE
            WRITE(IOUT,97) ((SLINE(J,I),J=1,2),I= 1, NUMHEAD)
            WRITE(IOUT,99) (DATA(11,I),I = 1 , NUMHEAD)
            WRITE(IOUT,1199) (DATA(12,I), I= 1 ,NUMHEAD)
            WRITE(IOUT,1299) (DATA(13,I), I= 1 ,NUMHEAD)
            DO  900  DECK = 1 , NUMHEAD
            NOO = DATA(11,DECK)
            DO 750 NN = 1 , NOO
            JJ = NN + 17
            CALL LSTCOE(DATA(JJ,DECK),ANS(1,NN,DECK) )
  750       CONTINUE
            IF(NOO.GE.98) GO TO  900
            NOO = NOO + 1
            DO 850 NN = NOO,98
            DO 850 I = 1,2
            ANS(I,NN,DECK) = BLANK
  850       CONTINUE
  900       CONTINUE
            DO 1000 NN = 1 , 98
            IF(NN.NE.1) GO TO 950
            WRITE(IOUT,1599)  NN , ( (ANS(I,NN,J),I=1,2 ), J= 1,NUMHEAD)
            GO TO 1000
  950       CONTINUE
            WRITE(IOUT,1699)  NN , ( (ANS(I,NN,J),I=1,2 ), J= 1, NUMHEAD)
 1000       CONTINUE
C   NOW LIST THE TAPE
            CALL OPSYS('LOAD','AMCMTAPE')
            CALL TAPSWC(LESSON,FLAG)
            IF(FLAG.EQ.0) GO TO 1200
            CALL ERROR(LNAME,0,2)
            RETURN
 1200       CONTINUE
C GET THE NUMBER OF STUDENTS
            READ(1,'I2') NSTUD
            IRLR = NUMHEAD
            DO 8000 STUDNT = 1, NSTUD
            NO = 0
 1500       CALL IFDATA(STUDNT, FLAG,LR=IRLR,COUNT,DATA)
            IF(FLAG.EQ.0) GO TO 2000
            IF(FLAG.EQ.1) GO TO 8000
            IF(FLAG.EQ.2) GO TO 9000
C TAPE ERROR
            CALL ERROR(LNAME,0,2)
            GO TO 9000
PAGE 03
```

FILMED FROM BEST AVAILABLE COPY


```
C WE'VE GOT SOME DATA TO LIST
 2000    CONTINUE
         NO = NO + 1
         IREX = IREX + COUNT
         IPAGE = IPAGE + 1
         CALL INFO(DATES)
         WRITE(IOUT,501) DATES , IPAGE
         I1 =((NO - 1) * 10) +  1
         I2 = I1 + COUNT - 1
         WRITE(IOUT,95) (I , I = I1 , I2  )
         WRITE(IOUT,88) (DATA(1,I) , I = 1 ,COUNT )
         IF(NO.NE.1 ) GO TO 2250
         READ(ISYNT'STUDENT) MASTE , NAME
 2250    CONTINUE
         WRITE(IOUT,90 )  NAME, STUDNT
         WRITE(IOUT,91 )  (DATA(3,I) , I = 1 , COUNT)
         WRITE(IOUT,92 )  (DATA(4,I) , I = 1 , COUNT)
         WRITE(IOUT,93 )  (DATA(5,I) , I = 1 , COUNT)
         WRITE(IOUT,96 )  (DATA(6,I) , I = 1 , COUNT)
         WRITE(IOUT,98 ) ((DATA(I,J), I = 7,9 ), J =1,COUNT)
         WRITE(IOUT,199)  (DATA(12,I),I = 1 , COUNT )
         DO 2600 I = 1 , COUNT
         DO 2500 J = 13, 17
         IF(DATA(J,I).NE.0) GO TO 2500
         DATA(J,I) = - 999
 2500    CONTINUE
 2600    CONTINUE
         WRITE(IOUT,139)  (  DATA(14,I),SLASH, DATA(15,J),SLASH
     2   , DATA(13,I) , I = 1 , COUNT )
         WRITE(IOUT,149)  (  DATA(16,I) , DOT , DATA(17,I), I = 1,COUNT)
         DO 4000  NOK = 1 , COUNT
         NOQ = 24 * DATA(6,NOK)
         DO 3500 NN = 1, NOQ
         JJ = NN + 17
         CALL LSTCOE(DATA(JJ,NOK ),ANS(1,NN,NOK ) )
 3500    CONTINUE
         IF(NOQ.GE.49) GO TO 4500
         NOQ = NOQ + 1
         DO 3800 NN = NOQ,48
         DO 3700 I = 1,2
         ANS(I,NN,NOK ) = BLANK
 3700    CONTINUE
 3800    CONTINUE
 4000    CONTINUE
 4500    CONTINUE
         DO 5000 NN = 1 , 48
         IF(NN.NE.1) GO TO 4850
         WRITE(IOUT,159) NN , ( ( ANS(I,NN,J) , I =1,2), J = 1,COUNT  )
         GO TO 5000
 4850    CONTINUE
         WRITE(IOUT,169) NN, ( ( ANS(I,NN,J) ,I =1,2), J = 1 , COUNT)
 5000    CONTINUE
         IF(LRI. EQ. 0) GO TO 8500
 8000    CONTINUE
 9000    IPAGE = IPAGE + 1
         CALL INFO(DATES)
         WRITE(IOUT,501) DATES, IPAGE
         WRITE(IOUT,9001) IREX, NOMEAD
 9001    FORMAT(1X, 'E3X , 'THERE ARE', I4,
```

FILMED FROM BEST AVAILABLE COPY


```
     2 'RECORDS '   ,        ', INCLUDING' , I4 , 1X , 'HEADERS'  )
    94     FORMAT(1X,'RECORD NUMBER',12X,10(3X,I2,4X) )
    95     FORMAT(1X,'DECK NUMBER',14X,10(3X,I2,4X) )
    88     FORMAT(1X,'LESSON',19X,10(3X,I2,4X)  )
    90     FORMAT(1X,'STUDENT NAME AND NUMBER',2X,13A2,2X,I3)
    91     FORMAT(1X,'SEGMENT',18X,10(3X,I2,4X) )
    92     FORMAT(1X,'TYPE',21X,   10(3X,I2,4X) )
    93     FORMAT(1X,'COURSE',19X,10(3X,I2,4X)  )
    96     FORMAT(1X,'NO. OF CARDS',13X,10(3X,I2,4X) )
    97     FORMAT(1X,'SINGLE OR DOUBLE',9X,10(1X,2A4) )
    98     FORMAT(1X,'IDENTIFICATION NUMBER',4X,10(2X,3A2,1X) )
    99     FORMAT(1X,'NO. OF SELECTIONS',8X,10(2X,I3,4X)  )
   109     FORMAT(1X,'NO. OF QUESTIONS ANSWERED',10(3X,I2,4X) )
   119     FORMAT(1X,'NO. OF QUESTIONS GRADED',2X,10(3X,I2,4X) )
   129     FORMAT(1X,'NO. OF QUESTIONS',9X,10(3X,I2,4X) )
   139     FORMAT(1X,'DATE MM/DD/YY',12X,10(1X,I2,A1,I2,A1,I2) )
   149     FORMAT(1X,'TIME HH.MM',15X,10(2X,I2,A1,I2) )
   159     FORMAT(1X,'QUESTION NUMBER',13,10X,10(1X,2A4) )
   169     FORMAT(1X,15X, I3,10X, 10(1X,2A4) )
    89     FORMAT(1X,'HEADERS' )
           END
/*
// LBLTYP NSD(5)
// EXEC LNKEDT
/&
// JOB 304138       REPTMAIN
// ASSGN SYSLNK,X'192'
// DLBL IJSYSLN,'SYSLNK',00/365,SD
// EXTENT SYSLNK,000002,1,0,10,1000
// OPTION CATAL
   PHASE REPTAIMS,ROOT
// EXEC FFORTRAN
C          XXXXXXXXXX     OUTPUT  GENERATOR          XXXXXXXXXX
C      FILE 5  -  LESSON SCRATCH FILE
C      FILE 6  -  HEADER FILE
C      FILE 7  -  DIRECTORY FILE
C      FILE 8  -  QUESTION FILE
C      FILE 9  -  STUDENT BACKGROUND FILE
C      FILE 10 -  STUDENT SCORE FILE
C      FILE 11 -  TEXT FILE
C      FILE  12 - SYSTEM FILE
       DEFINE FILE 5(1500,33,U,I5)
       DEFINE FILE 6(600,33,U,I6)
       DEFINE FILE 7(200,100,U,I7)
       DEFINE FILE 8(4000,35,U,I8)
       DEFINE FILE 9(250,33,U,I9)
       DEFINE FILE 10(8040,33,U,I10)
       DEFINE FILE 11(400,23,U,I11)
       DEFINE FILE 12(100,25,U,I12)
       INTEGER*2 IXP(16)
C      THE FLOW OF THIS PROGRAM IS AS FOLLOWS
C
C          1.   INITIALIZE
C          2.   INPUT MONITOR DATA
C          3.   SELECT A REPORT
C          4.   INPUT REPORT DATA
C          5.   ORGANIZE THE DATA FOR OUTPUT
C          6.   PRODUCE A REPORT
C          7.   SELECT NEXT REPORT
```

FILMED FROM BEST AVAILABLE COPY


```
C READ REPORT MONITOR DATA FORMAT IS FOR USER REPORTS ONE THRU 16
      READ(1,97) (IRP(K),K=1,16)
C THE FOLLOWING DECISION BLOCK IS DESIGNED
C TO DO THE FOLLOWING
C
C          1.   TEST WHICH REPORT NUMBERS
C               ARE  REQUESTED
C          2.   DIRECT THE PROGRAM TO THE SECTION
C               WHICH WILL PRODUCE THE REQUESTED
C               REPORT
C
C          IRP1 - IRP16 REPRESENTS REPORTS
C          ONE THRU SIXTEEN RESPECTIVELY
C
      IRESPT=4
      DO 100 KIRP=1,16
      REWIND IRESPT
      N=IRP(KIRP)
      GO TO (70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,85),N
      GO TO 100
C PRODUCE  REPORT  ONE
   70 CONTINUE
      CALL OPSYS('LOAD','REPT0001')
      CALL RSEP(IRP(KIRP))
      CALL REP1
      GO TO 100
C PRODUCE  REPORT  TWO
   71 CONTINUE
      CALL RSEP(IRP(KIRP))
      GO TO 100
C PRODUCE  REPORT  THREE
   72 CONTINUE
      CALL RSEP(IRP(KIRP))
      GO TO 100
C PRODUCE  REPORT  FOUR
   73 CONTINUE
      CALL OPSYS('LOAD','REPT0004')
      CALL RSEP(IRP(KIRP))
      CALL REP04
      GO TO 100
C PRODUCE  REPORT  FIVE
   74 CONTINUE
      CALL OPSYS('LOAD','REPT0004')
      CALL RSEP(IRP(KIRP))
      CALL REP05
      GO TO 100
C PRODUCE  REPORT  SIX
   75 CONTINUE
      CALL OPSYS('LOAD','REPT0004')
      CALL RSEP(IRP(KIRP))
      CALL REP06
      GO TO 100
C PRODUCE  REPORT SEVEN
   76 CONTINUE
      CALL OPSYS('LOAD','REPT0007')
      CALL RSEP(IRP(KIRP))
      CALL REP07
      GO TO 100
C PRODUCE  REPORT EIGHT
```

FILMED FROM BEST AVAILABLE COPY


```
   77 CONTINUE
      CALL RSEP(IRP(KIRP))
      GO TO 100
C PRODUCE  REPORT NINE
   78 CONTINUE
      CALL RSEP(IRP(KIRP))
      GO TO 100
C PRODUCE  REPORT TEN
   79 CONTINUE
      CALL RSEP(IRP(KIRP))
      GO TO 100
C  PRODUCE  REPORT ELEVEN
   80 CONTINUE
      CALL OPSYS('LOAD','REPT0012')
      CALL RSEP(IRP(KIRP))
      CALL REP11
      GO TO 100
C PRODUCE  REPORT TWELVE
   81 CONTINUE
      CALL OPSYS('LOAD','REPT0012')
      CALL RSEP(IRP(KIRP))
      CALL REP12
      GO TO 100
C PRODUCE  REPORT THIRTEEN
   82 CONTINUE
      CALL RSEP(IRP(KIRP))
      GO TO 100
C PRODUCE  REPORT FOURTEEN
   83 CONTINUE
      CALL RSEP(IRP(KIRP))
      CALL OPSYS('LOAD','REPT0013')
      CALL REP14
      GO TO 100
C PRODUCE  REPORT FIFTEEN
   84 CONTINUE
      CALL OPSYS('LOAD','REPT0015')
      CALL RSEP(IRP(KIRP))
      CALL REP15
      GO TO 100
   85 CONTINUE
      CALL OPSYS('LOAD','REPT0015')
      CALL RSEP(IRP(KIRP))
      CALL REP16
  100 CONTINUE
   65 WRITE(3,96)
   96 FORMAT(/,1X,'END OF AIMS OUTPUT GENERATOR')
   97 FORMAT(10X,16I2)
      CALL EXIT
      END
      SUBROUTINE HEADPG(IRPT,IPGE)
      INTEGER*2 INFORM(12),DATE(4),TIME(4)
      INTEGER*2 ORP(7),IRP
     1INTEGER D(10)/'****',' AIM','S OU','TPUT',' GEN','ERAT','OR. ',
     1'REPO','RT #','.  '/
      EQUIVALENCE (INFORM(5),DATE(1)),(INFORM(9),TIME(1))
      CALL INFO(INFORM)
      WRITE (3,10000) IRPT,TIME,DATE,IPGE
10000 FORMAT('1H1,//1H ,'    **** A.I.M.S. REPORT GENERATOR ****   REPORT
     2 NUMBER ',I2,' ****     TIME  - ',4A2,'    DATE  - ',4A2,'      PAGE
- PAGE 67
```

FILMED FROM BEST AVAILABLE COPY


```
     3 NUMBER ',I4,//)
       IPGE=IPGE+1
       RETURN
       ENTRY RSEP(IRP)
       IOTPT=3
       DO 10 K=1,7
    10 ORP(K)=IRP
       WRITE(IOTPT,20)
       DO 15 K=1,60
      WRITE(IOTPT,25) (ORP(J),J=1,7),(O(L),L=1,10),IRP,O(1),
      1(ORP(J),J=1,7),(O(L),L=1,10),IRP,O(1),(ORP(J),J=1,7)
    15 CONTINUE
    20 FORMAT(1H1)
    25 FORMAT(1X,2(7I2,10A4,I2,A4),7I2)
       RETURN
       END
       FUNCTION SUBGRD(IP,H,A)
       INTEGER*2 H,A,IP
       INTEGER*2 IPASS,SUBGRD,ITEST,N,D(11)
       INTEGER*2 THW(4)/3,5,7,9/,SCLEHW(4)/100,80,60,30/
       INTEGER*2 SCLESG(4)/100,80,40,10/
       SUBGRD=0
       ITEST=IPASS(H,A,N,D)
       GO TO (10,10,30,40,10),IP
       WRITE(IOTPT,500)
   500 FORMAT(' *** ERROR *** INCORRECT TYPE SPEC. ')
       RETURN
    10 CONTINUE
       IF(ITEST.EQ.1.AND.N.EQ.1) SUBGRD=100
       RETURN
    30 CONTINUE
       DO 32 K1=1,4
       IF(A.EQ. THW(K1)) SUBGRD=SCLEHW(K1)
    32 CONTINUE
       RETURN
    40 CONTINUE
       IF(ITEST.EQ.1) SUBGRD=SCLESG(N)
       RETURN
       END
       SUBROUTINE GETIT(RECORD,LESSON,SEGMNT,TYPE,FILES,ERRORS)
       INTEGER*2 LESSON,SEGMNT,TYPE,ERRORS,QUESTN, LPERM,CARD
       INTEGER*2 RECMBU(70),RECORD(65),RERORD(65),PTRS(50),NOS(50)
       INTEGER*2 WASTE(3)
       INTEGER SYSTEM,HEADER,QUESTO,TAPE
       INTEGER FILES(5)
       LOGICAL ONE
       SYSTEM=FILES(1)
       HEADER=FILES(2)
       QUESTO=FILES(3)
       TAPE=FILES(4)
       MIDDLE=1
       GO TO 1
       ENTRY GETREC(RECARD,LESSON,SEGMNT,TYPE,QUESTN,FILES,ERRORS)
       SYSTEM=FILES(1)
       HEADER=FILES(2)
       QUESTO=FILES(3)
       TAPE=FILES(4)
       MIDDLE=2
     1 READ (SYSTEM'2) LPERM,WASTE
```

FILMED FROM BEST AVAILABLE COPY


```
      READ (SYSTEM'3) PTRS
      READ (SYSTEM'4) NOS
      IF ((LESSON.GT.0).AND.(LESSON.LE.LPERM)) GO TO 2
      ERRORS=-1
      RETURN
    2 ISTART=PTRS(LESSON)
      IEND=ISTART+NOS(LESSON)-1
      DO 3 INDEX=ISTART,IEND
      READ (HEADER'INDEX) REKORD
      IF ((REKORD(3).EQ.SEGMNT ).AND.(REKORD(4).EQ.TYPE)) GO TO 4
    3 CONTINUE
      ERRORS=1
      RETURN
    4 ERRORS=0
      IF (MIDDLE.EQ.1) RETURN
      IPOINT=REKORD(8)+QUESTN-1
      IF (QUESTN.GT.REKORD(13)) GO TO 701
      READ(QUESTO'IPOINT) RECMBC
      RETURN
  701 ERRORS=-1
      RETURN
      ENTRY FINDIT(RECORD,LESSON,SEGMNT,TYPE,CARD,FILES,ERRORS,ONE)
      SYSTEM=FILES(1)
      HEADER=FILES(2)
      QUESTO=FILES(3)
      TAPE=FILES(4)
      ERRORS=0
      IF (.NOT.ONE) GO TO 100
      ONE=.FALSE.
      REWIND TAPE
  100 READ (TAPE,END=701) RECORD
      IF ((RECORD(1).EQ.LESSON).AND.(RECORD(3).EQ.SEGMNT).AND.(RECORD(4)
     2.EQ.TYPE)) GO TO 300
      IF((RECORD(1).GT.LESSON).AND.(CARD.EQ.2)) GO TO 701
      GO TO 100
  300 IF ((CARD.EQ.1).AND.(RECORD(2).EQ.0)) RETURN
      IF ((CARD.EQ.2).AND.(RECORD(2).NE.0)) RETURN
      IF (CARD.NE.1) GO TO 100
      BACKSPACE TAPE
      BACKSPACE TAPE
      GO TO 100
      END
      SUBROUTINE SUBMIT(TP,ST)
      IMPLICIT INTEGER*2 (K)
      LOGICAL*1 TEST(185,10)
      INTEGER*2 DID(12)/1,9,6,2,7,3,10,4,10,5,10,6/
      INTEGER*2 OUTCHK(10)
      INTEGER*2 TP(5),ST,SG
      INTEGER*2 CSN,RTP,RSG
      DO 10 K1=1,185
      DO 10 K2=1,10
      TEST(K1,K2)=.FALSE.
   10 CONTINUE
      RETURN
      ENTRY CHECK(CSN,RTP,RSG)
C IF TEST( ) IS TRUE THE STUDENT SUBMITTED MATERIAL
      TEST(CSN,DID(RTP*2*(RSG-1)))=.TRUE.
      RETURN
      ENTRY GETCHK(CSN,OUTCHK)
```

FILMED FROM BEST AVAILABLE COPY

```
        AIMS III SOURCE STATEMENT LISTING

      DO 20 K3=1,10
      OUTCHK(K3)=1
      IF(.NOT.TEST(CSR,K3)) OUTCHK(K3)=2
   20 CONTINUE
      OUTCHK(10)=1
      RETURN
      END
/*
// EXEC ASSEMBLY
IPASS    START
         CORE  ENTER
CORECT   EQU   4
STUDNT   EQU   6
ENTER    L     3,0(0,1)          1ST PAR IS CORECT ANS
         LH    5,0(0,3)          PLACE CORECT ANS IN GPR5
         L     3,4(0,1)          2ND PAR IS STUDENT'S RESPONSE
         LH    7,0(0,3)          PLACE STUDENT ANS IN GPR7
         NR    5,7               CHECK CORRECT BITS
         L     11,12(0,1)        LOAD GPR11 WITH ADDR OF COUNTS
*                                 COUNTS ARE 4TH PARAMETER
         USING CNTAR,11          TELL ASSEMBLER
         MVI   BLANK,X'00'       CLEAR COUNT AREA
         MVC   BLANK+1(21),BLANK
         SR    0,0               CLEAR GPR0 FOR TOTAL CORRECT
         SR    3,3               CLEAR GPR3 FOR TOTAL
         SLL   5,21              LEFT SHIFT TO SET UP FOR COUNTS
         SLL   7,21
*                                 THIS LOOP BEGINS THE COUNTING
*                                 SET UP FOR THE LOOP
         LA    10,18(0,0)        THIS IS OUR INDEX GPR
         L     8,=F'-2'          THIS IS THE INCREMENT
         L     9,=F'-1'          THIS IS THE COMPARAND
LOOP     SR    CORECT,CORECT     CLEAR THE HI ORDER GPR'S
         SR    STUDNT,STUDNT
         SLDL  CORECT,1          PLACE A BIT IN THE GPR
         SLDL  STUDNT,1          ALSO FOR HIS RESPONSE
         AR    0,CORECT          COUNT THE CORRECT BITS
         AR    3,STUDNT          COUNT THE STUDENT RESPONSE
         STM   STUDNT,COUNTS(10) AND COUNT THE RESPONSE
         BXH   10,8,LOOP         FINISH LOOP
         C     3,=F'0'
         BC    A'2',SOME         IF NONE OTHERS,THEN A BLANK
         L     4,=F'1'
         STH   4,BLANK           COUNT THE BLANK
SOME     L     4,8(0,1)          GET ADR OF 3RD PARAMETER
         STH   3,0(0,4)          PLACE RESPONSE CNT IN 3RD PAR
         MORE
CNTAR    DSECT
BLANK    DS    1H
COUNTS   DS    10H
         END
/*
   INCLUDE RIGHT
   INCLUDE INFO
   PHASE REPT0001,*
// EXEC FFORTRAN
      SUBROUTINE REP1
      DIMENSION ALINE(32)
      DIMENSION ONE(3),TWO(3),THR(3),FOU(3),FIVE(3)
```

FILMED FROM BEST AVAILABLE COPY


```
      INTEGER FILES(5)
      INTEGER*2 LESSON,SEGMNT,TYPE,ERRORS,QUESTN
      INTEGER * 2 RP,LN,ST,SC,CSN,T1,T2,T3,T4,T5
      INTEGER*2 RIGHT,TTEST,GRD,COUNT
      INTEGER * 2 HDR(48),ANS(48),RECM(65),RECR(65),RECM(65)
      INTEGER * 2 KEY(17),PRE(13)
      INTEGER * 2 TP(5),CARD,KTP,KST
      INTEGER * 2 CRNM(6)
      INTEGER * 2 GRD
      INTEGER * 2 NAMEND(23)
      INTEGER * 2 SECTN(2),GROUP(2),BACK(65)
      DATA WON/'PRE ','TEST','    '/,TWO/'POST',' TES','T   '/,
     1THR/'HOME','WORK','    '/,FOU/'STUD','Y GU','IDE '/,
     2FIV/'ASSI','GNME','NT  '/
      DATA ALINE/32*'----'/
      LOGICAL ONE
      EQUIVALENCE ( HDR(1), RECM(12)) , ( GRD , RECM(13))
      EQUIVALENCE ( KEY(1), RECR(41)) , ( PRE(1), RECM(58))
      EQUIVALENCE ( ANS(1), RECR(16)) , ( CSN , RECR(2)) , ( GRD , RECR(
     110))
      EQUIVALENCE (NAMEND(1) , BACK(1)) , (SECTN(1) , BACK(62)),
     1(GROUP(1) , BACK(64))
      READ(1,90) RP,LN,ST,SC,(TP(K),K=1,5),(CRNM(J),J=1,6)
   90 FORMAT(I2,7X,3I2,5I1,9X,6A2)
      LESSON = LN
      FILES(1) = 12
      FILES(2) =  6
      FILES(3) =  8
      FILES(4) =  4
      FILES(5) =  0
      GRAD = 0.00
      CARD = 2
      COUNT    =  0
      IRPT = 1
      IPGE = 1
      IOUT = 3
      ONE = .TRUE.
      DO 18 KTP=1,5
      IF(TP(KTP).LE.0.OR.TP(KTP).GT.5) GO TO 18
      TYPE= TP(KTP)
      SEGMNT=1
      IF(TP(KTP).NE.4) GO TO 17
      DO 16 KST=1,ST
      SEGMNT=KST
   17 CONTINUE
      CALL RSEP(RP)
      CALL GETIT (RECM,LESSON,SEGMNT,TYPE,FILES,ERRORS)
      IF(ERRORS.EQ.1) GO TO 12
      IF(ERRORS.EQ.-1)GO TO 12
    8 CALL FINDIT (RECR,LESSON,SEGMNT,TYPE,CARD,FILES,ERRORS,ONE)
C ERROR RETURN CHECK FOR FINDIT
      IF (ERRORS.EQ.-1) GO TO 10
C TITLES FOR REPORT ONE
      CALL HEADPG(IRPT,IPGE)
      INN = CSN
      READ (9'INN) (BACK(J),J=1,65)
      WRITE(3,100) (CRNM(J),J=1,6)
  100 FORMAT(40X,'STUDENT PERFORMANCE ANALYSIS FOR COURSE ',6A2)
      GO TO (26,27,28,29,30),TYPE
```

FILMED FROM BEST AVAILABLE COPY


```
   26 WRITE(IOUT,102) LN,(WON(J),J=1,3),SEGMNT
      GO TO 31
   27 WRITE(IOUT,102) LN,(TWO(J),J=1,3),SEGMNT
      GO TO 31
   28 WRITE(IOUT,102) LN,(THR(J),J=1,3),SEGMNT
      GO TO 31
   29 WRITE(IOUT,102) LN,(FOU(J),J=1,3),SEGMNT
      GO TO 31
   30 WRITE(IOUT,102) LN,(FIV(J),J=1,3),SEGMNT
  102 FORMAT(1HO,39X,'VOLUME ',I2,12X,3A4,13X,'SEGMENT ',I2)
   31 CONTINUE
      WRITE(IOUT,117) (ALINE(J),J=1,32)
  117 FORMAT(1X,32A4)
      WRITE(IOUT,115)
  115 FORMAT(1HO,1X,'CSN',16X,'NAME',16X,'ID NUMBER',22X,'SECTION',25X,
     1'GROUP')
      WRITE(IOUT,116) CSN,(NAMEWD(J),J=1,23),(SECTN(I),I=1,2),
     1(GROUP(K),K=1,2)
  116 FORMAT(1X,I4,23A2,22X,2A2,26X,2A2)
      WRITE(IOUT,117) (ALINE(J),J=1,32)
      WRITE(IOUT,103)
  103 FORMAT(1X,'QUESTION',7X,'ANSWER C/W',11X,'BEHAVIORAL OBJECTIVE',
     119X,'MESSAGE')
      DO 6 K = 1,GNO
      QUESTN = K
      ITEST = SUBGRD(TYPE,HDR(K),ANS(K))
      IF (ITEST.EQ.0) GO TO 7
      COUNT = COUNT+ ITEST
      WRITE(IOUT,107) K
  107 FORMAT(3X,I2,16X,'C')
      GO TO 6
    7 CALL GETMBO(RECN,LESSON,SEGMNT,TYPE,QUESTN,FILES,ERRORS)
      WRITE(IOUT,108) K,(KEY(J),J=1,17),(PRE(L),L=1,6)
  108 FORMAT(3X,I2,16X,'W',8X,17A2,6X,13A2)
    6 CONTINUE
      GRAD = COUNT/GNO
      IF(GRD.EQ.0) GO TO 21
      WRITE(IOUT,105) GRD
  105 FORMAT(1X,'YOUR GRADE IS  ',I3)
      WRITE(IOUT,104) GRAD
  104 FORMAT(1HO,1X,'YOUR RAW SCORE IS  ',F5.0)
      GO TO 22
   21 WRITE(IOUT,106) GRAD
  106 FORMAT(1HO,1X,'YOUR GRADE IS  ',F5.0)
   22 CONTINUE
      IF((COUNT/100+2).LT.GNO) WRITE(IOUT,999)
  999 FORMAT(1HO,1X,'YOU MUST SEE YOUR INSTRUCTOR THIS WEEK TO DISCUSS T
     1HIS TEST.')
      GRAD = 0.00
      COUNT = 0
      GO TO 9
C INSERT BEFORE RETURN
   12 WRITE(3,113) ERRORS
  113 FORMAT(1X,'GETIT ERROR LEVEL = ',I2)
      GO TO 11
   10 WRITE(3,112) ERRORS
  112 FORMAT(1H1,'FINDIT ERROR LEVEL = ',I2)
   98 FORMAT(1H1)
   11 CONTINUE
```

FILMED FROM BEST AVAILABLE COPY


```
      REWIND 4
      IF(TP(KTP).NE.4) GO TO 18
   16 CONTINUE
   18 CONTINUE
      RETURN
      END
/*
   PHASE REPT0004,REPT0001
// EXEC FFORTRAN
      SUBROUTINE REP04
      INTEGER * 2    IPASS,NP,D(11)
      INTEGER * 2    MARK/'**'/,BLANK/'  '/
      INTEGER * 2    GRAD,RLN,CSN,RSC,RTP,EC,LN,ST,SN,CTGT,EM,RP
      INTEGER * 2    CELL(40),STAR(40,20),TP(5),NAME(6)
      INTEGER * 2    HDRREC(65),RESREC(65),HDR(46),ANS(46),
     1               ERRORS,RIGHT,ITEST,COUNT,CARD
      INTEGER        FILES(5)/12,6,0,4,0/
      INTEGER * 2    GRC,CLWTH,CLWTH2,CELNUM(20),ZERO,MAXVAL,SCALE
      INTEGER * 2    INITL,ICNTL(20),INCR
      INTEGER  * 2 CHECK
      INTEGER  * 2      SCORES(40)
      INTEGER * 2  SCORE,SUBGRD
     CINTEGER     TYPNAM(3,5)/'PRE-','TEST','    ',
     1                        'POST',' TES','T   ',
     2                        'HOME',' WOR','K   ',
     3                        'STUD','Y GU','IDE ',
     4                        'ASSI','GNME','NT  ' /
      LOGICAL ONE
      EQUIVALENCE (HDR(1),HDRREC(18)),
     1            (ANS(1),RESREC(18)),
     2            (CSN   ,RESREC( 2)),
     3            (GRC   ,HDRREC(13))
      INPT  =  1
      IOUT  =  3
      CHECK  =  0
      CARD = 2
      IZERO = 0
      IRPT=4
      IPGE=1
      SCORE = 0.00
      ONE = .TRUE.
      CELL(1) = 0
      DO 98 K15=1,40
      DO 97 K14 = 1,20
      CELNUM(K14)=0
      SCORES(K15) = 0
      STAR(K15,K14) = 0
   97 CONTINUE
   98 CONTINUE
C
C READ IN REPORT CONTENT CARD
C     REQUIRED INFORMATION -
C              1 REPORT NUMBER    -  RP
C              2 LESSON NUMBER    -  LN
C              3 SEGMENT          -  ST
C              4 TYPE             -  TP
C              5 COURSE NAME      -  NAME
C
      READ(INPT,90) RP,LN,ST,SN,(TP(J),J=1,5),(NAME(I),I=1,6)
```

FILMED FROM BEST AVAILABLE COPY


```
   90 FORMAT(I2,7X,3I2,5I1,9X,6A2)
      DO 35 JJ = 1,5
      IF(TP(JJ).EQ.0) GO TO 99
C
C USE GETIT TO OBTAIN HEADER RECORD OF REQUIRED TEST
C
      CALL GETIT(HDRREC,LN,ST,TP,FILES,ERRORS)
      IF(ERRORS.EQ.1) GO TO 200
      IF(ERRORS.EQ.-1) GO TO 201
C
C COMPUTE CELL WIDTH AND CELL VALUES
C          SET UPPER CELL VALUE TO 100
C               THERE IS ONE CELL FOR EACH QUESTION IN THE TEST
C
      CLWTH = 100 / QNO
      DO 30 K4 = 1,QNO
      KK4 = K
      IF(K4.GT.1)  KK4 = K4  - 1
      CELL(K4) = CELL(KK4) + CLWTH
   30 CONTINUE
      CELL(QNO) = 100
C
C READ RESPONSE RECORD
C          GRADE STUDENT
C               ASSIGN GRADE TO CELL
C                    REPEAT UNTIL ALL STUDENTS ARE PROCESSED
C
  220 CALL FINDIT(RESREC,LN,ST,TP,CARD,FILES,ERRORS,ONE)
      IF(ERRORS.EQ.-1) GO TO 11
      COUNT = 0
      DO 210 K = 1,QNO
      ITEST = SUBGRD(TP,HDR(K),ANS(K))
      IF(ITEST.EQ.0) GO TO 210
      COUNT = COUNT+ ITEST
  210 CONTINUE
      SCORE = COUNT/QNO
      DO 40 K1 = 1,QNO
      IF(SCORE.NE.0) GO TO 42
      IZERO = IZERO + 1
      K1 = QNO
      GO TO 40
   42 IF(SCORE.GT.CELL(K1)) GO TO 40
      CELNUM(K1) = CELNUM(K1) + 1
      K1 = QNO
   40 CONTINUE
      GO TO 220
C
C SCALE CHART ACCORDING TO VALUES OF CELNUM
C          PRESENT MAXIMUM VALUES ARE - 40,80,120,160
C
   11 DO 61 K5 = 1,QNO
      IF (CELNUM(K5).GT.40) GO TO 60
   61 CONTINUE
      MAXVAL = 40
      SCALE  = 1
      GO TO 62
   60 DO 63 K6 = 1,QNO
      IF (CELNUM(K6).GT.80) GO TO 64
   63 CONTINUE
```

FILMED FROM BEST AVAILABLE COPY


```
      MAXVAL = 80
      SCALE  = 2
      GO TO 62
   64 DO 65 K7 = 1,GNO
      IF (CELNUM(K7).GT.120) GO TO 66
   65 CONTINUE
      MAXVAL = 120
      SCALE  = 3
      GO TO 62
   66 DO 67 K8 = 1,GNO
      IF (CELNUM(K8).GT.160) GO TO 68
   67 CONTINUE
      MAXVAL = 160
      SCALE  = 4
      GO TO 62
   68 WRITE(IOUT,100)
      WRITE(IOUT,107) (CELNUM(K11),K11=1,GNO)
  107 FORMAT(1H0,20I5)
  100 FORMAT(1H1,'ERROR , MAXVAL EXCEEDS LIMIT')
      RETURN
   62 CONTINUE
      DO 41 L1 = 1,40
      SCORES(L1) = (MAXVAL) - (L1 * SCALE) +  SCALE
   41 CONTINUE
      DO 70 K10 = 1,GNO
      DO 72 L2 = 1,40
      IF(CELNUM(K10).GE.SCORES(L2))  STAR(L2,K10) = MARK
   72 CONTINUE
   70 CONTINUE
C
C PRINT TITLES , MATRIX , STATISTICS
C
      CALL HEADPG(IAPI,IPGE)
      WRITE(IOUT,101) (NAME(I),I=1,8)
  101 FORMAT(1H0,28X,'AIMS HISTOGRAM ANALYSIS FOR ',8A2)
      WRITE(IOUT,92) LN,(TYPNAM(JJJ,IP(JJJ)),JJJ=1,3),ST
   92 FORMAT(1H0,20X,'VOLUME, ',I3,15X,'TYPE, ',3A4,15X,'SEGMENT, ',I3)
      WRITE(IOUT,300)
  300 FORMAT(20X ,72('-'))
      DO 80  K11 = 1,40
      WRITE (IOUT,103) SCORES(K11),(STAR(K11,K12),K12=1,GNO)
  103 FORMAT(I3,12X,20(A1,3X))
   80 CONTINUE
      WRITE(IOUT,104)
  104 FORMAT(2X,'0',71('-'))
      WRITE(IOUT,105)  (CELL(K13),K13= 1,GNO)
  105 FORMAT(' SCORES',8X,20(I3,1X))
      WRITE(IOUT,106) (CELNUM(K13),K13 = 1,GNO)
  106 FORMAT(' STUDENTS',4X,20(I3,1X))
      WRITE(IOUT,109) IZERO
  109 FORMAT(1H0,' NUMBER OF ZERO VALUES IN SAMPLE = ',I3)
   95 CONTINUE
      GO TO 96
  200 WRITE(IOUT,230)
  230 FORMAT(1H1,'GETIT = 1')
      GO TO 96
  201 WRITE(IOUT,231)
  231 FORMAT(1H1,'GETIT = -1')
      GO TO 96
```

FILMED FROM BEST AVAILABLE COPY

```
      ARMS III SOURCE STATEMENT LISTING

  202 WRITE(IOUT,232)
  232 FORMAT('1H1,'FINCIT = -1')
   90 RETURN
      END
      SUBROUTINE REPOS
C        COURSE STRUCTURE SUMMARY
      INTEGER*2 LESSON,SEGMNT,TYPE,CHAR,QUESTN,ID(2)
      INTEGER*2 DESC(17),PRESC(13),CHNM(6),RECM(70)
      INTEGER*2 CLES
      INTEGER * 2 ANSWER
      INTEGER * 2 A/'A'/,B/'B'/,C/'C'/,D/'D'/,E/'E'/
      EQUIVALENCE (LESSON , RECM(1)) ,(SEGMNT , RECM(2)),
     1(TYPE , RECM(3)) , (CHNO , RECM(4)) , (QUESTN , RECM(5)),
     2(ID(1), RECM(6)) , (DESC(1), RECM(8)), (PRESC(1), RECM(58))
     3,(ANSWER , RECM(12))
      IPGE=1
      IRPT=5
      IQUEST = 8
      IRPT   = 5
      IPGE   = 1
      IOUT   = 3
      INPT   = 1
      CHNO   = 0
      ILES   = 1
      CLES   = 0
      CNTR = 0
      READ(INPT,105) (CHNM(J),J=1,6)
  105 FORMAT(29X,6A2)
      CALL HEADPG(IRPT,IPGE)
   12 WRITE(IOUT,100)
  100 FORMAT(    40X,'COURSE STRUCTURE SUMMARY')
      READ(IQUEST'1) (RECM(J),J=1,70)
      WRITE(IOUT,101) CHNO,(CHNM(J),J=1,6)
  101 FORMAT('1H0,27X,'COURSE NO. ',I2,24X,'COURSE NAME  ',6A2)
   18 WRITE (IOUT,102)
  102 FORMAT('1H0,'VOLUME SEGMENT TYPE QUESTION  ID CD     DESCRIPTION
     1                    PRESCRIPTION',16X,'ANSWER')
   14 READ(IQUEST'ILES) (RECM(J),J=1,70)
      ILES = ILES + 1
      IF(ANSWER.EQ.3) ANSWER = A
      IF(ANSWER.EQ.5) ANSWER = B
      IF(ANSWER.EQ.9) ANSWER = C
      IF(ANSWER.EQ.17)ANSWER = D
      IF(ANSWER.EQ.33)ANSWER = E
      IF(LESSON.EQ.0) GO TO 20
      WRITE(IOUT,103) LESSON,SEGMNT ,TYPE,QUESTN,(ID(J),J=1,2),
     1(DESC(J),J=1,17),(PRESC(J),J=1,13)
     2,ANSWER
  103 FORMAT(2X,I2,6X,I2,4X,I2,5X,I2,5X,2I3,3X,17A2,13A2,4X,A4)
      CNTR = CNTR + 1
      IF(CNTR.EQ.45) GO TO 16
      GO TO 14
   16 CNTR = 0
      CALL HEADPG(IRPT,IPGE)
      GO TO 18
   20 WRITE(IOUT,104)
  104 FORMAT('1H0,40X,'END OF PROCESSING')
      RETURN
      END
```

FILMED FROM BEST AVAILABLE COPY

```
C     ***  ITEM ANALYSIS FOR AIPS VERSION 2  ***
      INTEGER FILES(5)
      INTEGER ATAPE
      INTEGER * 2  NAME(40),ANSWER(40),LESSON,SEGMNT,TYPE,TYPES(5)
      INTEGER*2 CRS(40,10),ICRS(40),ICRSS(10),NUMREC,SCORE,CRSNAM(6)
      INTEGER*2 STAN(40,10),CARDS,CARD,MARK,BLANK,ONE,IFAIL,NUMCOR
      INTEGER*2 NUMREC(65),NUM(40),IDENT,COMMEN,MEAN,ANSFOR(10)
      INTEGER*2 ALPOUT(10)/'A','B','C','D','E','F','G','H','I','J'/
      INTEGER*2 COMMNT/'C'/,INCANS,LNTAPE,CNTTST,TOTAL
      INTEGER*2 TOTSEL,INCANS,NUMSEL
      INTEGER   CASH(10)/10*'-----'/
      CHARACTER     ITEMANL(5)//'ITE-','ITEM','     ',
     1                          'PCSB','ES  ','I   ',
     2                          'ALME','AN  ','K   ',
     3                          'STND','.DEV','ERR ',
     4                          'ASSI','GNPE','NT  '/
      EQUIVALENCE (ICRS(1),NUMREC(1))
      EQUIVALENCE (ANSWER(1),ALPOUT(1))
      EQUIVALENCE(NUM(1),NUMREC(2)) , (CNT , NUMREC(3))
      DATA MARK/'**'/,BLANK/'  '/
      LOGICAL UNF
      INPT    = 5
      IPUN    = 1
      IOUT    = 3
      INPI    = 1
      ATAPE   = 4
      CARD = 2
      FILES(1)  =   12
      FILES(2)  =    6
      FILES(3)  =    8
      FILES(4)  =    9
      FILES(5)  =    5
      ONE = .TRUE.
C     ***   INITIALIZE  MATRICES   ***
      DO 5 I=1,40
      ICRS(I) =0
      DO 6 J=1,10
      ICRSS(J) =0
      CRS(I,J)  =  0
      STAN(I,J) =  BLANK
    6 CONTINUE
    5 CONTINUE
      TOTAL = 0
      NUMBER = 0
      CNTMUL=0
      CNTTST=0
C     ***   READ  IN  REPORT  CONTENT  CARD   ***
      READ(INPT,90) REPNO,LESSON,SEGMAT,SECTA,(TYPES(K),K=1,5),
     1(CRSNAM(J),J=1,6)
   90 FORMAT(I2,7X,3I2,5I1,9X,6A2)
C     ***   MAJOR  LOOP  IS  FOR  TYPE  NUMBER   ***
      DO 34 K=1,5
      TYPE = TYPES(K)
      IF(TYPES(K).EQ.0) GO TO 30
C     ***
C     ***   INSERT  ASTERISK  FOR  CORRECT  ANSWER   ***
      CALL GETIT(NUMREC,LESSON,SEGMNT,TYPE,FILES,ERRORS)
      NUMSEL = TOTSEL/ONE
```

FILMED FROM BEST AVAILABLE COPY

```
      [illegible]

      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
  [illegible] CONTINUE
      [illegible]
C     ***
C     ***   [illegible]   ***
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
C     ***   [illegible]   ***
C     ***
C     ***   [illegible]   ***
      [illegible]
      [illegible]
C     ***   FILL IN [illegible]   ***
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
  [illegible] CONTINUE
      [illegible]
      [illegible]
C     ***
C     ***   COMPUTE LINE TOTALS   ***
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
  [illegible] FORMAT([illegible])
      [illegible]
C     ***
      [illegible]
  [illegible] FORMAT(10X,'TOTAL',7X,10(A3,4X))
      [illegible]
  [illegible] FORMAT(10X,'SELECTIONS ',10(I4,3X),3X,I5)
      [illegible]
PAGE 78
```

FILMED FROM BEST AVAILABLE COPY

```
```

FILMED FROM BEST AVAILABLE COPY


```
C
      WRITE (IPF,14)
   14 FORMAT(1H ,40X,'TERMINAL COLLECTIVE IMMEDIATE LEVEL SUMMARY'//40X,
     2'----------------------------------------'//1H ,30X,'TERMINAL CO
     3LLECTIVE',/34X,'IMMEDIATE LEVEL',/30X,'--------------------',/33X,'----
     4----------',//1H0)
C
C         THAT READS THE PAGE
C
   15 READ (IRCD,10,END=100) INDEX,ILEV
   16 FORMAT (2I3)
      IF (ILEVELS(INDEX).GT.0) GO TO 7002
      ILEVELS(INDEX)=ILEV
      GO TO 15
 7002 WRITE(IPF,7003) INDEX,ILEV,ILEVELS(INDEX)
 7003 FORMAT (1H0,'**** ERROR **** DUPLICATE TERMINAL SPECIFICATION-
     2  I=',I3,' ATTEMPTED VALUE = ',I3,' ORIGINAL VALUE = ',I3,///
     3/////)
      CALL EXIT
  100 N=0
      DO 102 K=1,200
      IF(ILEVELS(K).EQ.00000) GO TO 102
      N=N+1
      WRITE (IPF,101) K,ILEVELS(K)
  101 FORMAT(1H ,3I8,I3,25X,I3)
      IF(N.LE.44) GO TO 102
      CALL HEADPG(IPF,PAGE)
      N=0
      WRITE(IPF,14)
  102 CONTINUE
C         SU PGRM FOR GETTING AND LISTING LEVELS
      READ (IRDR,1) NCOMM
      DATA ISTOR/05*0/
      IF(NCS.NOSTEP+2
      DO 3 NCOMM=1,NFUNS
      WRITE(ISCM,ICOMM)ISTOR
    3 CONTINUE
      READ (IRDR,2) LPCMP,NPCMP,LICMP,NICMP,ICMPFG
      READ (IRDR,3) PIMS
      READ (IRDR,4) NOS
      ISTART=PEAS(INLES)
      IEND=ISTART+NUSE(INLES)-1
      IDIFF=ISTART-1
      IF ((INLES.GT.LPCOM).OR.(INLES.LT.0)) CALL ERRT
   4  DO 2 IMP=ISTART,IEND
      INDEX=IMP-IDIFF
      READ (IMEAD,IMP) (MDRS(K,INDEX),K=1,60)
      OCTOLE(INDEX,1)=MDRS(4,INDEX)
      OCTOLE(INDEX,2)=MDRS(5,INDEX)
      NGUES=MDRS(13,INDEX)
      IPOINT=MDRS(8,INDEX)
      IAMP=IPOINT+NGUES-1
      DO 2 J=IPOINT,IAMP
      READ (IQUES'J) RECQ
      INDX1=J-IPOINT+1
      TOTOLE(INDX1,INDEX)=RECQ(IC)
      ITC=RECQ(IC)
      IOCNTS(ITC)=IOCNTS(ITC)+1
    2 CONTINUE
```

FILMED FROM BEST AVAILABLE COPY

```
      [illegible] SOURCE STATEMENT LISTING

      REWIND ITAPE
   11 READ (ITAPE,END=7000) RECS
      IF (RECS(1).NE.ISLES) GO TO 12
      IF (RECS(2).EQ.0) GO TO 11
      GO TO 13
   12 IF (RECS(1).GT.ISLES) GO TO 7000
      GO TO 11
 7000 WRITE (IPT,7001) ISLES
 7001 FORMAT (1H1,'NO RECORD OF LESSON ',I3,///////)
      CALL EXIT
   13 CONTINUE
C
C               AT THIS POINT, I'VE GOT  1 THE FIRST MIN
C                     ICTOLL  -          2 A TABLE OF TU'S VS. CONSTRUCTS
C                     LCTOLS             3 A TABLE OF LOCKS VS. TYPES
C                     IUCNTS             4 I
C                     IUCPTS             5 A TABLE OF NO. OF CONSTR'S PER
C                     MORE               6 ALL THE PROGRAMS FOR A LESSON
C   NOW SET TIME TO BE THE REAL WORK
      I=5 GO 100 IN=1,200
      I=6 STUDREC(IN)=0000
      I=4 ITYPE=REC5(4)
          ISCO=REC5(3)
          GO 107 JN=1,10
          IF(ICTOL(1,1,JN).NE.ITYPE) GO TO 107
          IF(ICTOL(1,2,JN).LE.ISCO) GO TO 108
  107 CONTINUE
          WRITE (IPT,7004) ITYPE,ICCTOL
 7004 FORMAT(1H1,'ILLEGAL TYPE NUMBER= ',I2,/1H0,'MAXIMUM TYPES ARE =',
         2I2,1H.//////)
          CALL EXIT
  108 IUSER=IN
          NCOUR=MOD(REC5(1),1000)
          GO 110 IN=1,MAXSTU
          IF (MAXSTU*17.LT.IUSER).EQ.1) GO TO 110
          IAV=IN+17
          IF (ABS(FLOAT(REC5(IAV)-NCOUR)/FLOAT(IUSER)).GT.1.E-06) GO TO 110
          ITOND=TOTAL(IUSER)
          STUDREC(ITOND)=STUDREC(ITOND)+1
  110 CONTINUE
          ISAVE=REC5(2)
          READ(ITAPE,END=10000) RECS
          IF (REC5(2).EQ.ISAVE) GO TO 104
C         IF IT GETS HERE, I'M DONE WITH THE REC
          DO 111 IN=1,200
          ITOP=STUDREC(IN)
          ICOI=TOTAL(IN)
          IF(IBOT .EQ. 0) GO TO 111
          STUDREC(IN)= (100*ITOP)/IBOT
          IF (STUDREC(IN).LE.LEVEL(ITYP)) GO TO 111
          STUDREC(200)=1
          IICS(IN)=IICS(IN)+1
  111 CONTINUE
          IF(STUDREC(200).GT.0) DATA(3)=DATA(3)+1
          IF(STUDREC(200).EQ.0) GO TO 120
          CALL HEADPG(IPT,PAGE)
          READ(IBKGD'ISAVE)RECKG
          WRITE (IPT,112) (RECKG(IAT),IAT=0,80),(RECKG(IAT),IAT=19,23),IMRCS
  112 FORMAT  (1H0,13A2,' LOG NUMBER  ',5A2,/1H ,'THE STUDENT IDENTIFIC
PAGE 61
```

FILMED FROM BEST AVAILABLE COPY

```
      AIMS III SOURCE STATEMENT LISTING

      SO ABOVE HAS PERFORMED BELOW PRE-SET LEVELS OF PERFORMANCE ON TERMI
     1NAL OBJECTIVE(S) IN LESSON *,I2,/1HO,*----------------------------
     2---------- PERFORMANCE DATA ----------------------------------------
     3---*,/1HO,9A,*I,3A,*,20A,*PERCENT AND*,2OA,*CUTOFF LEVEL*,/1HO)
       DO 111 I=1,264
       IF(ISTUFC(I)).LT. 00000) GO TO 111
       WRITE (IPI,114)I,STUFC(I),LEVELS(I)
  114 FORMAT (1P ,5X,I3,2X,I3,3X,I3)
  111 CONTINUE
  120 IPCENT=(ISAVE*100)
      ICHAR=IPCENT
      INDEX=ICHAR
      ISA=INDEX
      WRITE (IOUT,*)IPCENT,ISTU1
      WRITE (IOUT,*)INDEX,ISTU2
      WRITE(IOUT,*)ISTU3,ISTU3
      WRITE(IOUT,*)ISTU4,ISTU4
      IF (ISTU3(I).GE.ISTU3(J)) GO TO 8000
      GO TO 1C5
 8000 DATA2B=ISTU3(I)
      DATA1B=0
      DO 8001 I=1,200
      IF (ISTU3(I).LT.0) GO TO 8001
      DATA1B=DATA1B+1
 8001 CONTINUE
      DO 8100 I=1,1+1
      IF(ISTUFC(I).LT. 00000) GO TO 8100
      IF(ISTUFC(I) .LT. 00000) GO TO 8100
      CALL HEADPG(IPI,PAGE)
      WRITE(IPI,90023) INL,IMO,ALGNT,IEND,IBEGIN,LEVELS(I)
 9002 FORMAT (1HO,40X,*TERMINAL OBJECTIVE REPORTING SUMMARY*,/1HO,20A,*LE
     2SSON *,I2,3A,*TERMINAL OBJECTIVE *,I3,4A,*COMPOSED OF *,I2,* QUEST
     3IONS.*,//1OX,*A TOTAL OF *,I3,* STUDENTS PERFORMED BELOW THE CUTOFF
     4 LEVEL SET AT *,I3,//1OX,*THE FOLLOWING STUDENTS HAVE PERFORMED BELO
     5W THE CUTOFF LEVEL-*,//11X,*NAME OF STUDENT*,12X,*SCORE NO.*,3X,*PER
     6CENT CHANGE*,//1X,20I*-*,2X,10I*-*,5X,*----------*,///
      DO 9005 I=1,MISTER
      IPCENT=(ISAVE*40-3
      INEXT=IPCENT
      ISAV=INEXT
      IMX=INEXT
      READ(ISCM*IPCENT)ISTU1
      READ(ISCM*INEXT)ISTU2
      READ(ISCM*IMX) ISTU3
      READ(ISCM*IMX)ISTU4
      IF(ISTUFC(I).LT. LEVELS(I)) GO TO 9005
      READ (IDSKGD*IMX)IBEGIN
      WRITE(IPI,90004) (LOCNAME(K),IMX=1,232 ) , STUNEC(I)
 9004 FORMAT(I3MO,6X,I3A2,2X,I3A2,3X,I3)
 9005 CONTINUE
 8100 CONTINUE
      RETURN
      END
      SUBROUTINE DEVICE (IDATA,ITOS,INLES,LEVELS)
C
C THIS PROGRAM WILL ASSIGN THE IL'S TO BE TAUGHT IN THE RESPECTIVE
C   ATAFORIAL SESSIONS. IT CAN HANDLE UP TO 40 SESSIONS AND 99 IO'S.
C
      DIMENSION IGATE(0),IVERE(200,2),LOCNAME(20),ILEASS(10,20),IDATA(5)
PAGE 02
```

FILMED FROM BEST AVAILABLE COPY

```
      WITH THE SOURCE STATEMENT LISTING

      DIMENSION ITOS(200),LEVEL(200),LOC(20),ITEMP(20)
      INTEGER PAGE
      INTEGER*2 INFORM(12),DATE(4),TIME(4),RECORD(80)
      EQUIVALENCE (INFORM(5),DATE(1)),(INFORM(9),TIME(1))
      COMMON /FILES/ RCD,IP1,IP2,IP7,IP8,ISCR,INEAC,IOUT,IWORK,ISTUD,ISCORE,
     1LIBRAT,ISYS,IP3,IP4,IP5
      DATA LOC/20*'----'/
      PAGE =1
      INPT=5

C
C  READ IN THE NUMBER OF AEROBICAL SESSIONS AND THE MAXIMUM NUMBER
C    OF STUDENTS ALLOWED IN A SESSION.
C
      READ(ICD,3)NGMI,MSTUD
    3 FORMAT(2I3)
C
C  READ IN CLASS LOCATION AND PROFESSOR CARDS.
C
      DO 3 I=1,10
      DO 3 J=1,20
      LOC(I,J)=LOC(J)
    3 CONTINUE
      I=0
    9 READ(ICD,2,END=6) ITEMP
    2 FORMAT(20A4)
      I=I+1
      DO 4 J=1,20
    4 LOC(I,J)=ITEMP(J)
      GO TO 9
    6 IF(I.LT.NGMI) WRITE(IP3,7)
    7 FORMAT(1H0,'******  ERROR - NUMBER OF GMI SESSIONS EXCEEDS THE NUM
     1BER OF LOCATION DESCRIPTION CARDS***')
C
C  SORT ID'S FROM MOST TO LEAST MISSED.
C
      DO 5 I=1,200
      IVEX(1,1)=I
    5 IVEX(1,2)=ITOS(I)
      DO 10 I=1,199
      JI=I+1
      DO 10 J=JI,200
      IF (IVEX(I,2).GE.IVEX(J,2)) GO TO 10
      DO 8 K=1,2
      ICAT(K)=IVEX(I,K)
      IVEX(I,K)=IVEX(J,K)
    8 IVEX(J,K)=ICAT(K)
   10 CONTINUE
C
C  CHECK THAT NO. OF I.D.'S NOT LESS THAN NO. OF SESSIONS
C
      IF(IDATA(1).LT.NGMI) NGMI=IDATA(1)
C
C  DETERMINE HOW MANY ID'S ARE TO BE ASSIGNED TO EACH GMI SESSION.
C
      I=IDATA(1)/NGMI
      II=I*NGMI
      IREM=IDATA(1)-II
C
C  SET UP NUMBER OF ID'S IN EACH SESSION
PAGE 63
```

FILMED FROM BEST AVAILABLE COPY


```
C
      DO 20 J=1,NUMI
      K=NUMI-J+1
      IF (IDATA(K)-ID(K)) 11,11,12
   12 ICLASS(K)=I+1
      ANEW=IDATA(K)-1
      GO TO 20
   11 ICLASS(K)=I
   20 CONTINUE
C
C  DETERMINE WHICH ILP'S ARE TO BE ASSIGNED TO EACH SESSION.
C
      IDIFF=IDATA(II)-II
      KK=0
      ICNT=1
      DO 30 I=1,NUMI
      IK=IK+1
      IF(IK.GE.IDATA(I)) GO TO 30
      ICLASS(IP,ICNT)=IK
   25 CONTINUE
      ICNT=ICNT+1
      IF(ICNT.LT.4) GO TO 30
      IK=NUMI-IDIFF+1
      DO 31 IP=IP,NUMI
      IK=IK+1
      IF(IK.GE.IDATA(II)) GO TO 39
      ICLASS(IP,ICNT)=IK
   31 CONTINUE
   39 CONTINUE
C
C  PRINT OUT CLASS ASSIGNMENT OF ILP'S
C
C
      DO 50 I=1,NUMI
      CALL READPG(IRPT,PAGE)
      WRITE(IPT,33) (ALES,I,(LOC(I,J),J=1,20)
   33 FORMAT(1HO,///43X,'REMEDIAL SESSION - LESSON PLAN - LESSON ',I3///2
     1X,'SESSION ',I3,5X,'FACP. AND/OR LOCATION  ',1X,20A4///32X,'PROGRAM
     2E OBJECTIVE',1X,'CUTOFF LEVEL',3X,'NO. OF STUDENTS'//32X,10(I2,'-',I
     3A,12('-'),2X,A5('-')//)
      K1=ICNT(I)
      DO 40 J=1,K1
      WRITE(IPT,45)(WORK(ICLASS(I,J),I),LEVELS(ICLASS(I,J)),IVLAE
     1(ICLASS(I,J),2)
   45 FORMAT(1HO,34(I4,2X,A4,2X),I4,18X,I5)
   40 CONTINUE
   50 CONTINUE
      CALL CONVER(ALES,IDATA,IIDS,LEVELS,ICLASS,IVEX,NUMI,NSTUD,ICNT,LO
     1CC)
      RETURN
      END
      SUBROUTINE CONVER (ALES,IDATA,IIDS,LEVELS,ICLASS,IVEX,NUMI,NSTUD,
     1ICNT,LOC)
C
C  THIS ROUTINE WILL ASSIGN STUDENTS TO THE PROPER REMEDIAL SESSION
C
      DIMENSION ICNT(10),IVEX(200,2),LOC(10,20),ICLASS(10,20),IDATA(5)
      DIMENSION IIDS(200),LEVELS(200),NCLAS(10,100),IPOINT(90)
      INTEGER*2 INFORM(12),DATE(4),TIME(4),RECD(65),STDATA(201),ICTL(11)
```

FILMED FROM BEST AVAILABLE COPY

```
      WIMS III SOURCE STATEMENT LISTING

      EQUIVALENCE (INFORM(5),CATE(1)),(INFORM(6),TIME(1))
      COMMON/FILES/ICC,IP1,IP2,IP3,IP4,IP5,IP6,IP7,IP8,IP9,IP10,IP11,I
     CIP12,IP13,IP14,IP15,IP16,IP17
C
C  CREATE RANDOM SEED AND NUMBER TO START SEARCH OF STUDENT FILE
C  IN ORDER TO ASSURE FAIRNESS IN CHOICE OF SECTION
C
C
      IMPI=13
      IPAGE=1
      CALL IMPCINFO(INF)
      ISEED=DATE(2)*DATE(3)+TIME(4)
      ISEED=IABS(ISEED)
      ISEED=(ISEED/63)*2+1
      CALL RANDU(ISEED,IY,YFL)
C
C  CALCULATE STARTING STUDENT NUMBER
C
      ENUM=YFL*IDATA(2)
      ENUM=ENUM-1
C
C  ZERO OUT THE CLASS LOCATORS AND STUDENT NUMBERS IN CLASS
C
      DO 5 I=1,NUMI
      DO 4 J=1,NSTUD
    4 MCLASS(I,J)=0
    5 ICT(I)=0
      N=NCHI+1
      ICT(N)=0
C
C  READ IN STUDENTS FROM FILE 5 WITH IMSIM NO'S
C
      IRT=IDATA(2)-ENUM+1
      KI=IDATA(2)
      DO 100 IRCD=1,KI
      IF (IR.EQ.IRT) GO TO 10
      IV=IV+1
      END=EV
      GO TO 11
   10 IV=0
      END=ENUM+(IRCD-1)
   11 JSC=(4*IND)-3
      READ(ISCH'JSC)(STDATA(J),J=1,80)
      JSC=JSC+1
      READ(ISCH'JSC)(STDATA(J),J=81,160)
      JSC=JSC+1
      READ(ISCH'JSC)(STDATA(J),J=161,195)
      JSC=JSC+1
      READ(ISCH'JSC)(STDATA(J),J=196,201)
      IF(STDATA(201).EQ.0)GO TO 100
C
C  DETERMINE WHICH SECTION THE STUDENT BELONGS IN BY FINDING TO GROUP
C  HE HAS MISSED MOST ON
C
      ISAVE=0
      ISUM=0
      DO 50 I=1,NCHI
      JI=ICHI(I)
      ISUM=0
```

FILMED FROM BEST AVAILABLE COPY

```
      ARMS III SOURCE STATEMENT LISTING

      DO 40 J=1,JI
      A=IVEA(ICLASS(I,J),I)
      ISUM=ISUM+IDATA(A)
   40 CONTINUE
      IF(ISUM.LT.ISAM) GO TO 30
      ISAM=ISUM
      ISAVE=I
      GO TO 30
   30 IF(ISUM.LT.ISAM) GO TO 50
      ISAM=ISUM
      ISAVE=I
   50 CONTINUE
C
C  CHECK THAT CLASS IS NOT FULL
C
      IF(ICTC(ISAVE).GE.NSTUD)GO TO 60
      ICTC(ISAVE)=ICTC(ISAVE)+1
      NCLASS(ISAVE,ICTC(ISAVE))=IND
      GO TO 100
C
C  IF FULL TRY SECOND CHOICE
C
   60 IF(ICTC(ISAVE).GE.NSTUD)GO TO 70
      ICTC(ISAVE)=ICTC(ISAVE)+1
      NCLASS(ISAVE,ICTC(ISAVE))=IND
      GO TO 100
C
C  SET UP FILE OF DOUBLE REJECTIONS
C
   70 NREJ=NREJ+1
      ICTC(N)=ICTC(N)+1
      IREJ(ICTC(N))=IND
  100 CONTINUE
C
C  PRINT OUT CLASS ROSTERS
C
      DO 200 I=1,NUMI
      CALL HEADPG(IPT,IPAGE)
      II=ICH(I)
      WRITE(IPT,110)(LEVELS(IVEA(ICLASS(I,J),I),J=1,I
     C1)
  110 FORMAT(1HO,43X,'REMEDIAL SESSION ROSTER'//44X,23('-')//47X,'LESSON
     C',I3/47X,10('-')//47X,'SESSION ',I3/47X,11('-')//20X,'PROF AND/OR
     C EDUCATION- ',20A4///05X,'TERMINAL OBJECTIVES'//40X,40('-')//10X,'TER
     CMINAL OBJECTIVE NUMBERS-   ',20I4)
      WRITE(IPT,109) (LEVELS(IVEX(ICLASS(I,J),1)),J=1,II)
  109 FORMAT(1HO,9X,'THRESHOLD LEVELS SET AT   -   ',20I4)
      WRITE(IPT,111)
  111 FORMAT(1HO,'THE FOLLOWING STUDENTS HAVE BEEN ASSIGNED TO THIS REME
     CDIAL SESSION'///1X,'STUDENT NAME',10X,'NUMBER',15X,'PERCENTAGE ERRO
     CR ON THE ABOVE TERMINAL OBJECTIVES'/1X,26('-'),2X,6('-'),5X,60('-'
     C))
      A=ICTC(I)
      IF(A.GT.0) GO TO 312
      WRITE(IPT,220)
  220 FORMAT(1HO,10X,'****    NO STUDENTS ASSIGNED TO THIS SESSION    **
     1***)
```

FILMED FROM BEST AVAILABLE COPY

```
      ###2 ### SOURCE STATEMENT LISTING

          GO TO 200
      312 DO 350 IW=1,K
      310 CONTINUE
          READ(1STUD'#CLASS(I,IW))RECD
      300 CONTINUE
          IND=NCLASS(I,IW)
          IST=64*(NCL)-1
          READ(1SCM'1ST)(STUDATA(J),J=1,65)
      305 CONTINUE
          IST=IST+1
          READ(1SCM'1ST)(STUDATA(J),J=66,130)
      307 CONTINUE
          IST=IST+1
          READ(1SCM'1ST)(STUDATA(J),J=131,195)
          CONTINUE
          IST=IST+1
          READ(1SCM'1ST)(STUDATA(J),J=196,200)
          CONTINUE
          KT=NCNT(I)
      311 CONTINUE
          WRITE(IPT,215)(RECD(J),J=6,18),NCLAS (I,IW),(STUDATA(WEA(CLASS(I,
         CJ),I)),J=1,KT)
      215 FORMAT(1H ,13A2,3A,I5,5A,2014)
      350 CONTINUE
      200 CONTINUE
          NPAGE=1
          K=NCLASS(N)
          IF(K.LT.1) GO TO 280
          CALL HEADPG(IPT,NPAGE)
          WRITE(IPT,210)NALES
      210 FORMAT(1HO,43X,'REMEDIAL SESSION ROSTER'//44X,23('-')///47X,'LESSON-
         C ',I3//47X,11('-')///20X,     'THE FOLLOWING STUDENTS HAVE NOT BEEN
         CNTTED ENG IMEIR FIRST OR SECOND CHOICE'///10X,'ASSIGNMENT TO A
         CSESSION WILL HAVE TO BE DONE BY THE PROFESSOR.'///)
          WRITE(IPT,205)
      205 FORMAT(1H ,15X,'STUDENT NAME',12X,'NUMBER',5X,'SESSION ASSIGNMENT'
         C/8X,20('-'),5X,5('-'),5X,20('-')///)
          DO 250 I=1,K
          KI=IREJCT(I)
          READ(1STUD'KI) RECD
          WRITE(IPT,211)(RECD(J),J=6,18),IREJCT(I)
      211 FORMAT(1H ,7A,13A2,5A,I5)
      250 CONTINUE
      280 RETURN
          END
/*
   INCLUDE RANDU
   PHASE REPTOOL2,REPTOLO1
// EXEC FFORTRAN
      SUBROUTINE REPII
      IMPLICIT INTEGER*2(K)
      INTEGER*2 CHNK(6),IP(5),KP,LN,ST,SC,NIMFIL,NINCRS
      INTEGER*2 RESREC(65),NVOL,NCSN,NSG,NTP
      INTEGER*2 STUDCK(65),OUTCHK(10),STNAME(15)
      INTEGER IOTPT/3/,INPTC/1/,IRESPT/4/,IALMSY/12/,ISTUDK/9/
      INTEGER OUTPUT(2)/'SUB.','NO/S'/
      INTEGER*2 CSN
      EQUIVALENCE (NVOL, RESREC(1)) , (NCSN,RESREC(2)),(NSG,RESREC(3))
      EQUIVALENCE (NTP,RESREC(4)),(STNAME(1),STDUCK(6)),(CSN,STUDCK(2))
PAGE 87
```

FILMED FROM BEST AVAILABLE COPY

```
      AIMS III SOURCE STATEMENT LISTING

      EQUIVALENCE (ALARGP, STORGE(24))
      IPGE=1
      IOPT=11
      READ(IAIMSY'1) NAMFIL,NRECRS
      READ(IMPIC,10) PP,LA,ST,SC,(IP(K),K=1,5),(CRAM(K),K=1,6)
   10 FORMAT(12,7X,312,5X,2X,9X,5X2)
 2070 CONTINUE
      CALL SUCHEK(IP,SB)
      ERRNO=0
      KSTU=1
 2010 READ(IAESPI,END=2000) AESREC
      IF(LN-KWDE) 2030,2020,2010
 2020 CONTINUE
      CALL CHECK(MESS,IP,KSC)
      GO TO 2010
 2030 ERRNO=-1
      BACKSPACE IAESPI
 2000 CONTINUE
      KPGE= (NENFIL/45)+1
      DO 3000 KOZ=1,KPGE
      CALL HEADPG(IOPT,IPGE)
      WRITE(IOTPT,510)
  510 FORMAT  (55Z,'*** VOLUME SUBMITTAL REVIEW ***')
      WRITE(IOTPT,515) (CRAM(K),K=1,6),LA
  515 FORMAT( /,40A,'COURSE: ',6A2,5A,'VOLUME NO.  ',I3)
      WRITE(IOTPT,550)
  550 FORMAT(/,24X'NOTE:  NO/S - SIGNIFIES THAT STUDENT DID NOT SUBMIT M
     1ATERIALS FOR PROCESSING',
     2/,31X'SUB. - SIGNIFIES THAT STUDENT DID SUBMIT MATERIALS FOR PROCE
     3SSING',/)
      WRITE(IOTPT,530)
  530FORMAT  (10X,'STUDENT',15X,'CSN  PAR.',5(3A,'STUDY'),
     1'  ASSIGN    NAME    POS#',/,
     237X,'TEST   QUE(1)  QUE(2)  QUE(3)  QUE(4)  QUE(5)',
     311X,'SCORE    TEST')
      DO 3010 KOJ=1,45
      READ(ISTOBK'KSTU)STOBCK
      IF(KDROP.EQ..)GO TO 1030
      CALL GETCHK(CSN,OUTCHK)
      WRITE(IOTPT,520) (STONME(K1),K1=1,13),CSN,(OUTPUT(OUTCHK(K2)),
     1K2=1,9)
      GO TO 1070
 1030 WRITE(IOTPT,540)(STONME(K1),K1=1,13),CSN
 1070 IF(KSTU.GE.NINFIL) GO TO 3000
      KSTU=KSTU+1
 3010 CONTINUE
 3000 CONTINUE
      IF(ERRNO.EQ.0) GO TO 9999
      ENDFILE
      GO TO 2070
  520 FORMAT(1X,12A2,A1,6X     ,I3,2X,9(A4,4X))
  540 FORMAT(1X,12A2,A1,6X,I3,'   STUDENT DROPPED')
 9999 RETURN
      END
      SUBROUTINE REPI2
      INTEGER*2 STOBCK(65),STONME(13),CRAM(6),NINFIL,NRECRS
      EQUIVALENCE (STONME(1),STOBCK(6))
      DATA INPIC,IOTP,ISTOB,IAIMSY,IPGE/1,3,9,12,1/
      JSTU=1
```

FILMED FROM BEST AVAILABLE COPY

```
        AIMS III SOURCE STATEMENT LISTING

        READ(INPIC,100)RP,(CHAR(K),K=1,6)
        READ(IAIMSY'II)NINFIL,NINCRS
        ACUM2=(NINFIL/45)+1
        DO 1000 ACUM1=1,ACUM2
        CALL HEADPG(RPT,IPGE)
        WRITE(IOTPT,520)(CHAR(K),K=1,6)
        WRITE(IOTPT,530)
        DO 1100 ACUM3=1,45
        READ(ISTOR'JSTO)STOCK
        WRITE(IOTPT,540)(STONME(K2),K2=1,13),(STOCK(K3),K3=19,23),STOCK(
       12)
        IF(JSTO.EQ.NINFIL) GO TO 1000
        JSTO=JSTO+1
 1100 CONTINUE
 1000 CONTINUE
   100 FORMAT(12,2A,6A2)
   510 FORMAT(1H,12A2,A1,5X,5A2,5A,I4)
   520 FORMAT(55X,'FOR CLASS ADVISER ***',//,35X,'COURSE ',6A2,//)
   530 FORMAT(10X,'STUDENT',14X,'ACADEMY',8X,'COURSE',/,33X,'NO.',10X,
      1'STD.NO.')
       RETURN
       END
/*
   PHASE REPT0013,REPTCODE
// EXEC FFCATRAN
       SUBROUTINE REP14
       IMPLICIT INTEGER*2 (K)
       INTEGER*2 RLN,ERRG
       INTEGER*2 NINFIL,NINCRS
       INTEGER*2 CHAR(6),TP(5),IC
       INTEGER*2 RP,LN,ST,SC,CTGT,CH,JSTO,ACUMP,TESTCT,STONO
       INTEGER*2 STONME(13),OUTPUT(15)
       EQUIVALENCE (JSTO,STONO)
       INTEGER INPIC/1/,IOTPT/3/,IRESPT/4/,IAIMSY/12/
       OUTC=0
       JSTO=1
       IRPT=14
       IPGE=1
       KIL=0
C                                                    SRCNIN,LIBNIN )1'YSMIAI(DATA
       NINFIL=10
       NINCRS=10
       READ(INPIC,10) RP,LN,ST,SC,(TP(J),J=1,5),(CHNP(J1),J1=1,6)
 2020 CONTINUE
       CALL MEDREC(RP,LN,ST,IP,CTGT,CH)
       IF(CH.NE.0) GO TO 999
       CALL SUBMIT(RP,ST)
       CALL CUNAVL
       CALL GETREC(RP,LN,RLN,NINFIL,NINCRS,ERRG)
       CALL CRPST
       CALL REPIS(CHAR)
       JOUR2=(NINFIL/45)+1
       DO 1100 JOUR1=1,JOUR2
       CALL HEADPG(IRPT,IPGE)
       WRITE(IOTPT,510)
       WRITE(IOTPT,515)(CHAR(K),K=1,6),LN
       WRITE(IOTPT,501)
       WRITE(3,530)
       DO 1000 JOUR3=1,45
```

FILMED FROM BEST AVAILABLE COPY

```
      [illegible] SOURCE STATEMENT [illegible]

      [illegible]
      [illegible]
      [illegible]
```

FILMED FROM BEST AVAILABLE COPY

```
C     SEE IF SOURCE STATEMENT LISTING
      WRITE(6,2000) (SN(I),CUMSLE(I),CUMALE(I),A=1,4)
      RETURN
      ENTRY OUTCUM(LS,CRAP)
      WRITE(6,2020) (CRAP(K),K=1,LS)
 2000 FORMAT(' NUMBER ACCUMULATED FOR CUM. ABS. ERROR IN COLUMN ',F8.2,
     1' WILLBE NO. ',I3)
      DO 3000 NF=1,NF
      READ(5,2001) (SN(I),CUMSLE(I),CUMALE(I),A=1,4)
      WRITE(6,2020) (CUMSLE(I),CUMALE(I),A=1,4)
 3000 CONTINUE
      RETURN
      END
      SUBROUTINE CRAP(NF,LA,NLA,NF,MX,CIGF,CD)
      DIMENSION NP,NF,NC,CIGI,LA
      DIMENSION DS(185,11),UC(175)
      DIMENSION GRAD,NLA,CSA,ASD,MIP,IC
      DIMENSION OUTCMX(100),KSI
      DIMENSION CRAM(6)
      DATA(DS(I,J),I=1,5,10,11/
      DO 4000 NI=1,NF
      CC(KSI)=0
      DO 4000 KI=1,11
      CC(KSI,KI2)=0
 4000 CONTINUE
 2000 CONTINUE
      CALL GRAM(GRAD,NLA,CSA,ASD,MIP,CD)
      IF(CD.NE.0) GO TO 2010
      CALL CHECK(CSA,MIP,ASD)
      ENDFILE(7P+24(ASD-1))
      CC(5,6,L)=GRAD
      GO TO 2000
 2010 CONTINUE
      DO 2200 KSI=1,NF
      CALL GET.MK(KSI,CLICMK)
      DO 2300 KI=1,9
      IF(OUTCMX(KI).EQ.0) CC(KSI)=CC(KSI)+1
 2300 CONTINUE
      IF(CC(KSI).NE.CIGI.AND.CC(KSI).NE.0) CC(KSI)=-2
 2200 CONTINUE
      GO FORMAT(11,2016)
      RETURN
      DATA# GAPSI
C     VARIABLY ALLOCATION FOR GROUP STATISTICS
      INTEGER   GRPSIT(7,40),20(7)
      INTEGER*2 STDARE(165,13),STDOCK(6),SN(13),CAPIA,STOND
      EQUIVALENCE (STOND,STDOCK(2)),(CAPIA,STDOCK(2)),(SA,STDOCK(6))
      NI=1
      NGRPC=0
      DO 3000 NI=1,7
      GRPSIT(NI,1)= 1000
      GRPSIT(NI,2)= 0
      GRPSIT(NI,3)=-1000
      GRPSIT(NI,4)= 0
 3000 CONTINUE
      IOCA=0
        DO 2111 A9=1,NF
        READ(IOCK,K,STDOCK
        IF(STDOCK(24).EQ.1) CC(STDNO)=-1
        IF(CAPIA.EQ.-999) CC(STDNO)=-2
```

FILMED FROM BEST AVAILABLE COPY

```
      ARMS III SOURCE STATEMENT LISTING

      DIMENSION=11=CARIN
      DO 2120 K=1,13
      STORE(DIMENSION,K)= SAVE(K)
 2120 CONTINUE
 2130 CONTINUE
      DO 2100 KSTD=1,NF
      IF(DS(KSTD,1)-2.0*DS(KSTD,2).LE.0) GO TO 2100
      DS(KSTD,2) = (NI*DS(KSTD,1)+DS(KSTD,2))/(NI+1)
      DS(KSTD,7)= (DS(KSTD,3)+DS(KSTD,5))/2
      DS(KSTD,3)=DS(KSTD,2)-DS(KSTD,1)
      DS(KSTD,8)=DS(KSTD,7)-DS(KSTD,1)
      DO 3010 NGI=1,7
      AD(NGI)=DS(KSTD,NGI)/I
 3010 CONTINUE
      DO 3020 NGI=1,7
      GRPSTI(NGI,1)=AMIN1(GRPSTI(NGI,1),AD(NGI))
      GRPSTI(NGI,3)=AMAX1(GRPSTI(NGI,3),AD(NGI))
      GRPSTI(NGI,4)= GRPSTI(NGI,4)+AD(NGI)
 3020 CONTINUE
      NGRPC=NGRPC+1
 2100 CONTINUE
      IF(NGRPC.EQ.0) GO TO 3030
      DO 3040 NGI=1,7
      GRPSTI(NGI,2) = GRPSTI(NGI,4)/NGRPC
 3030 CONTINUE
      GRPSTI(3,2)= GRPSTI(2,2)-GRPSTI(1,2)
      GRPSTI(4,2)= GRPSTI(7,2)-GRPSTI(1,2)
      DO 3040 KSTD=1,NF
      DS(KSTD,4)=DS(KSTD,3)-GRPSTI(3,2)
      DS(KSTD,9)=DS(KSTD,8)-GRPSTI(4,2)
 3040 CONTINUE
 3050 RETURN
      ENTRY HEADPG(IRPI,IPAG)
      IRPI=1
      IPAG=1
      IOUTPI=3
      CALL HEADPG(IRPI,IPAG)
      WRITE(IOUTPI,5100)(CRAM(K),K=1,6),LA
 5100 FORMAT(20X,'*** VOLUME STATISTICS ***',//,10X,'COURSE ',A2,
     1 10X,'VOLUME NO.  ',I3,//,23X,'MINIMUM',7X,'MEAN',8X,'MAXIMUM')
      WRITE(IOUTPI,5200) (GRPSTI(1,KG),KG=1,3),
     1                   (GRPSTI(2,KG),KG=1,3),
     1                    GRPSTI(3,2),
     1                   (GRPSTI(5,KG),KG=1,3),
     1                   (GRPSTI(6,KG),KG=1,3),
     1                    GRPSTI(7,2),
     1                    GRPSTI(4,2)
 5200 FORMAT(1X,'CAPABILITY INDEX',          I2,3(3X,I3),/,
     1       1X,'PERFORMANCE INDEX',         I2,3(3X,I3),/,
     2       1X,'PERFORMANCE DEVIATION',     I3,3X,/,
     3        1X'PROBLEM ACHIEVEMENT'        I2,3(3X,I3),/,
     4        1X'POST TEST ACHIEVEMENT',     I2,3(3X,I3),/,
     5       1X,'NET ACHIEVEMENT INDEX',     I3,3X,/,
     6       1X,'ACHIEVEMENT DEVIATION',     I3,I3)
      RETURN
      ENTRY GETRES(JSTD,SM,OUTPUT,NGRCP, IESTC1)
      INTEGER*2 OUTPUT(13),JSTD,KGRCP,IESTCT,SM(13)
      DO 4000 KI=1,9
      OUTPUT(KI)=DS(JSTD,KI)
```

FILMED FROM BEST AVAILABLE COPY

```
      ANSI III SOURCE STATEMENT LISTING

  4000 CONTINUE
       DO 5010 KI=1,15
       SM(KI)=STUDNT(DSTD,KI)
  5010 CONTINUE
       KOUNT=0
       IF(CC(JSTD).EQ.-1) KOUNT=1
       RESTC=CC(JSTD)
       RETURN
       END
       SUBROUTINE GRADE(GRAD,KLN,CSA,RSG,RIP,EG)
C THIS SECTION OF SUBR GRADE READS A RESP. RECORD SEQUENTIALLY
C AND GRADES IMC TEST
       IMPLICIT INTEGER*2 (K)
       INTEGER FILES(5)/12,0,3,-6,7,IMCSPI/-7,1GIPI/3/
       INTEGER*2 MORC(12,49),MK(65),RESPR(65),ANS(48),NUMQI(12),IP(5)
       INTEGER*2 GRAD,CSA,RSG,RIP,EG,KLN,ST,CTL,I,M,KLN
       INTEGER*2 COUNT,IPE,KNO,MIGMI,IIEST,SG,SM
       INTEGER*2 SUBGRD,SUMGRD,NUMASP(12),MORCD(49)
       EQUIVALENCE (RESPR(1),ANS(1)),(KNO,MK(1))
       EG=0
       GRAD=0
       COUNT=0
C READ RESPONSE TAPE
  5000 READ(IRESPT,END=9999)RESPR
       KLN=RESPR(1)
C TEST FOR PROPER VOL. NO. (KLN)
       IF(KLN-KLN) 5020,5010,5000
  5010 CONTINUE
       CSA=RESPR(2)
       RSG=RESPR(3)
       RIP=RESPR(4)
C THIS EQUATION MAPS A 2-D FIELD AND A 1-D FIELD
       IPE= RIP*2+RSG-1
       KNO=NUMQI(IPE)
C CALCULATE A NUMERIC GRADE FOR TEST
       DO 4000 K=1,KNO
C FUNCTION SUBGRD GRADES RESP. USING APPROP. KEYS
       IIEST= SUBGRD(RIP,MORC(IPE,K),ANS(K))
       COUNT=COUNT+IIEST
  4000 CONTINUE
       GRAD = COUNT/KNO
       IF(RIP.EQ.1.OR.RIP.NE.4) GO TO 999
       DO 3030 K=1,KNO
       MORCD(K)=MORC(IPE,K)
  3030 CONTINUE
       GRAD = SUMGRD(KNO,NUMASP(IPE),MORCD,ANS)
   999 RETURN
C EG=1 IS RETURN CODE FOR END OF FILE
  9999 EG=1
       RETURN
  5020 CONTINUE
C EG=-1 RETURN CODE DIFFERENT VOL. NO.
       EG=-1
       KLN=KLN
       BACKSPACE  IRESPT
       RETURN
C
C
       ENTRY MORKEY(RIP,KLN,ST,IP,CTL,I,M)
```

FILMED FROM BEST AVAILABLE COPY

```
      AIMS III SOURCE STATEMENT LESSIMW

      WRITE(6,IPI,>15)
C
C THIS SECTION PICKS OUT THE MEDIMS REQUIRED TO GRADE THE TEST
C AND STORES THEM IN THE ARRAY CALLED  MDPC
C
      DO 2060 KZ=1,12
      NUMG(KZ)=1
      DO 2060 K3=1,45
      MDPC(KZ,K3)=0
 2060 CONTINUE
      CIQT=0
C THIS LOOP PICKS MEDIM HEADER RECORD AND STORES IT
C     PLUS COUNTS TOTAL EXAMS TO BE GRADED (CIQT)
C     AND NUMBER OF QUESTIONS PER TEST (NUMQT)
      DO 1500 K1=1,5
      SG=1
      IF(IP(K1).EQ.0.OR.IP(K1).GT.5) GO TO 1500
      IF(IP(K1).NE.4) GO TO 1530
 1520 DO 1540 SR=1,ST
      SG=SR
 1530 CALL GET(IEM,LN,SG,IP(K1),FILES,EM)
      IF(EM.NE.0) WRITE(6,IPT,>1G) EM,LN,IP(K1),SG
      IPE=IP(K1)*2*(SG-1)
      DO 1550 K2=1,40
      MDMG(IPE,K2)= MM(17,K2)
 1550 CONTINUE
      CIQT=CIQT+1
      NUMQT(IPE)=QNO
      NUMRSP(IPE)=MM(11)
      IF(IP(K1).NE.4) GO TO 1500
 1540 CONTINUE
 1500 CONTINUE
      RETURN
  >1G FORMAT(' *** WARNING *** ERROR IN OBTAINING MEADER RECORD,  ERROR,
     1',I2,' LESSON.',I2,' TYPE.',I2,' SEGMENT.',I2)
  >15 FORMAT(1H1)
      END
      FUNCTION SUMGRD(Q,N,M,A)
      INTEGER*2 Q,M(48),A(48),N,K
      INTEGER*2 IPASS,SUMGRD,ICORR,NP,O(11),SCORR,SNP,INCORR
   10 FORMAT(40I3)
      SCORR=0
      SNP=0
      DO 1000 K=1,Q
      ICORR=IPASS(M(K),A(K),NP,O)
      SCORR=SCORR+ICORR
      SNP=SNP+NP
 1000 CONTINUE
      INCORR=SNP-SCORR
      SUMGRD= (SCORR*100-(Q*INCORR*100 / (N-1)))/Q
      RETURN
      END
/*
   PHASE REPIOOIS,REPICOOI
// EXEC PROGRAM
      SUBROUTINE REPIS
      IMPLICIT INTEGER*2 (A)
      INTEGER*2 NINFIL,NINCRS
      INTEGER*2 OUTPUT(185,16),FILE(185,2),MAIL(185,2),CUTCAN(165)
PAGE 34
```

FILMED FROM BEST AVAILABLE COPY

```
      ARMS III SOURCE STATEMENT LISTING

      INTEGER*2 STUDCK(65),STNME(13),CRNM(6),IP(5)
      INTEGER*2 OUTCTR(10),CTRI,ANS,NTI,CAPIN
      INTEGER*2 RP,LN,SI,SC,GRAD,RLN,CSN,RSG,RIP,EG,EM,IND,CTQI,LEN
      INTEGER*2 SECT(2),GROUP(2),STUNO,CUPAV
      INTEGER*2 SUBOUT(2)/'  ','***'/
      INTEGER*2 KSTD,PTDUM(2),MKDUM(2),ERRCUM
      INTEGER*2 DIG(12)/1,9,8,2,7,3,10,4,20,5,10,0/
      INTEGER INPIC/1/,IOTPT/3/,IRDSPT/4/,IAIMST/12/
     EQUIVALENCE (STUNME(1),STUDCK(6)),(STUNO,STUDCK(2))
      EQUIVALENCE (KDUMCP,STUDCK(24)),(CAPIN,STUDCK(25))
      EQUIVALENCE (SECT(1),STUDCK(42)),(GROUP(1),STUDCK(44))
      IRPT=15
C READ IN THE NO. OF STUDENTS ENROLLED IN THE COURSE.
      READ(IAIMST'1) NINFIL,NINCRS
C READ IN THE REPORT REQUEST CARD
      READ(INPTC,10) RP,LN,SI,SC,(IP(J),J=1,5),(CRNM(J1),J1=1,6),IGRPT
      IF(LN.EQ.1) CALL CURSES(NINFIL)
C THIS LOOP RETRIEVES CURR. AVG. DATA FROM SCORE FILE
      DO 3010 KSTD = 1,NINFIL
      CALL GETCRA(KSTD,LN,PTDUM,MKDUM,ERRCUM)
      IF (ERRCUM.EQ.-1) GO TO 4000
      DO 3010 L1=1,2
      PTLI(KSTD,L1) = PTDUM(L1)
      MKLI(KSTD,L1) = MKDUM(L1)
 3010 CONTINUE
 2070 CONTINUE
      CALL RSEP(RP)
      JSTD=1
      IPGE=1
C ZERO ARRAYS AS REQUIRED.
      DO 2020 KDUM7=1,NINFIL
      OUTCTR(KDUM7)=0
      DO 2020 KDUM8=1,10
      OUTPUT(KDUM7,KDUM8)=0
 2020 CONTINUE
C SET UP HEADER ARRAY
      CALL HEDRTI(LN,SI,IP,CTQI,EM)
C SET UP FOR STUDENT SUBMITTAL CHECK
      CALL SUBMIT(IP,SI)
      KREAD=NINFIL*CTQI
C THIS LOOP READS IN STUDENT RESPONSE TAPE,
C     ALSO GRADES THE TEST AND STORES RESULT (OUTPUT(CSN,IND).
      DO 2000 KDUM4=1,KREAD
      CALL GRADI(GRAD,RLN,CSN,RSG,RIP,EG)
      IF(EG.NE.0) GO TO 2010
C CHECK ENTRY POINT IN SUBMIT, HERE WE INDICATE THAT STUD, HAS HANDED IN
      CALL CHECK(CSN,RIP,RSG)
C MAP FUNCTION FOR OUTPUT ARRAY
      IND=DIG(RIP*2*(RSG-1))
      OUTPUT(CSN,IND)=GRAD
      OUTCTR(CSN)=OUTCTR(CSN)+1
 2000 CONTINUE
 2010 CONTINUE
      JDUM2=(NINFIL/45)+1
C THIS BEGINS THE LOOP FOR THE OUTPUT OF GRADES
      DO 1100 JDUM1=1,JDUM2
      CALL HEADPG(IRPT,IPGE)
      WRITE(IOTPT,510)
      WRITE(IOTPT,515)(CRNM(K2),K2=1,6),LN
PAGE 95
```

FILMED FROM BEST AVAILABLE COPY

```
      ARPS III SOURCE STATEMENT LISTING

      WRITE(ICIPI,501)
      WRITE(ICIPI,550)
      DO 1000 JSTU=1,65
C READ STUDENT BACKGROUND FILE
      READ(9'JSTU)STUBK
      IF(NGROUP.EQ.1) GO TO 1030
      IF(OUTCHA(STUNO).LT.0) GO TO 1040
C FIND OUT WHAT THE STUDENT WAS SUBJECTED FOR PROCESSING
      CALL OUTCHK(STUNO,OUTCHA)
      CIPI=0
      SUMAVE=0
C CALCULATE THE SUM OF ALL GRADE
      DO 1060 KAVE=2,8
      IF(OUTCHA(KAVE).EQ.1) CIPI=CIPI+1
C CALCULATE PREF. INDEX (SUMAVE)
      SUMAVE=SUMAVE+OUTPUT(STUNO,KAVE)
 1060 CONTINUE
      IF(CIPI.EQ.0) CIPI=1
      OUTPUT(STUNO,10)=SUMAVE/CIPI
      PTLT(STUNO,1)= PTLT(STUNO,1)+OUTPUT(STUNO,9)
      MMLT(STUNO,1)= MMLT(STUNO,1)+OUTPUT(STUNO,8)
C CALCULATE THE REQUIRED AVERAGES
      CUMAV=0
      NPT=1
      NMM=1
      IF(OUTCHA(8).EQ.2) NMM=0
      IF(OUTCHA(9).EQ.2) NPT=0
      MMLT(STUNO,2)=MMLT(STUNO,2)+NMM
      PTLT(STUNO,2)=PTLT(STUNO,2)+NPT
C CALCULATE CUM.AVE. DELO=
      DIF(MMLT(STUNO,2).EQ.0.AND.PTLT(STUNO,2).GT.0)
      1CUMAV =       PTLT(STUNO,1)/PTLT(STUNO,2)
      DIF(PTLT(STUNO,2).EQ.0.AND.MMLT(STUNO,2).GT.0)
      1CUMAV =       MMLT(STUNO,1)/MMLT(STUNO,2)
       IF(MMLT(STUNO,2).GT.0.AND.PTLT(STUNO,2).GT.0)
      1CUMAV =  3*MMLT(STUNO,1)/(MMLT(STUNO,2)*10)+
      2          7*PTLT(STUNO,1)/(PTLT(STUNO,2)*10)
C OUTPUT NORMAL LINE OF STATISTICS
      WRITE(ICIPI,520) (STUNME(K4),K4=1,13),SECT(1),SECT(2),GROP(2),
      2STUNO,
      1(OUTPUT(STUNO,K4),SUBOUT(OUTCHA(K4)),K4=1,10),CUMAV,CAPIN
      DIF(IGRPT.NE.0)
      WRITE(IGRPT,521)LN,(STUNME(K),K=1,13),SECT(1),SECT(2),GROP(2),
      2STUNO,(OUTPUT(STUNO,K4),SUBOUT(OUTCHA(K4)),K4=1,10),CUMAV,
      3SUBOUT(1),CAPIN,SUBOUT(1)
       GO TO 1070
C OUTPUT STUD. DROPPED LINE
 1030 WRITE(ICIPI,540)(STUNME(K5),K5=1,13),STUNO
      GO TO 1070
 1040 WRITE(ICIPI,550)(STUNME(K5),K5=1,13),STUNO
 1070 IF(JSTU.GE.MINFIL) GO TO 1100
      JSTU=JSTU+1
 1000 CONTINUE
 1100 CONTINUE
      IF(EG) 2040,2040,2060
 2040 CONTINUE
      LN=LN
      GO TO 2070
 2060 CONTINUE
```

FILMED FROM BEST AVAILABLE COPY

```
      WAPS III SOURCE STATEMENT LISTING

C THIS ENTRY LOADS UPDATED CUM. AVG. DATA BACK ON SCORE FILE
      DO 3020 KSTU=1,NINFIL
      DO 3030 L1=1,2
      PTOUM(L1) = PILT(KSTU,L1)
      MMOUM(L1) = MMLT(KSTU,L1)
 3030 CONTINUE
      CALL LOCCUM(KSTU,LN,PTOUM,MMOUM)
 3020 CONTINUE
      IF(IGRPT.NE.0) ESU FILE IGRPI
      GO TO 999
 4000 WRITE(ILIPT,5000) LN,LA
 5000FORMAT(' *** ERROR *** VOLUME NO.',I3,' HAS BEEN PROCESSED BY REP
     1ORT 13. REPORT REQUEST TERMINATED.',//,20X,'TO GENERATE REPORT IS
     2FOR VOLUME NO.',I3,' REQUEST A REPORT FOR VOLUME ONE')
   10 FORMAT(I2,7X,3I2,5I1,9X,6A2,2X,I2)
   13 FORMAT(62X,I2)
   14 FORMAT(16(I5,I3))
  501 FORMAT(1H0)
  510 FORMAT(//,50X,'*** VOLUME SUMMARY ***')
  515 FORMAT(///,40X,'COURSE ',6A2,3X,'VOLUME NO.  ',I3,//,20X,'NOTE,
     1THE ** SIGNIFIES THAT NO MATERIALS HAVE BEEN SUBMITTED FOR PROCESS
     1ING.')
  520 FORMAT(1A,12A2,A1,3A2     ,I3,16(2X,I3,'.',A2),2X,I3,'.',A1,I3,'.')
  521 FORMAT(62,12A2,A1,3A2,I3,12(I3,'.',A2))
 5300FORMAT  (10X,'STUDENT',15X,'CSN  PRE.',5(3X,'STUDY'),
     1'  ASSIGN     MORE     POST    PERF.     CUM.    CAPE.',//,
     237A,'TEST    GDE(1)  GDE(2)  GDE(3)  GDE(4)  GDE(5)',
     311A,'NORM    TEST    INDEX    AVE.    INDEX')
  540 FORMAT(1X,12A2,A1,9X,I3,9A,'STUDENT DROPPED')
  550 FORMAT(1X,12A2,A1,9A,I3,6A,'THIS STUDENT DOES NOT HAVE ANY MATERIA
     1LS TO BE PROCESSED')
  999 RETURN
      END
      SUBROUTINE CONDIS(NINFIL)
      INTEGER*2 NINFIL,ZERO/0/,CSN,LN,ERRCUM,VOL
      INTEGER*2 PILT(2),MMLT(2),DUMMY(65)
      ISCORE=10
C THIS SECTION ZERO'S THE SCORE FILE FOR FRESH START
      DO 10 JSTU=1,NINFIL
      CSN= JSTU/1
      READ(ISCORE'JSTU) DUMMY
      WRITE(ISCORE'JSTU)  CSN,ZERO,ZERO,ZERO,ZERO,ZERO,(DUMMY(I),I=7,65)
   10 CONTINUE
      RETURN
C THIS ENTRY FINDS CUM. AVERAGE DATA
C     ALSO CHECKS FOR OVER LAPPING REPORT REQUEST
      ENTRY GETCUM(CSN,LN,PILT,MMLT,ERRCUM)
      ISCORE=10
      JCSN=CSN/1
      READ(ISCORE'JCSN)  CSN,VOL,PILT,MMLT
      ERRCUM=0
      IF(VOL.GE.LN) ERRCUN=-1
      RETURN
C THIS ENTRY LOADS UPDATED CUM. AVERAGE DATA BACK ON FILE
      ENTRY LOCCUM(CSN,LN,PILT,MMLT)
      JCSN = CSN/1
      READ(ISCORE'JCSN) DUMMY
      WRITE(ISCORE'JCSN) CSN,LN,PILT,MMLT,(DUMMY(I),I=7,65)
      RETURN
```

FILMED FROM BEST AVAILABLE COPY


```
      END
      SUBROUTINE GRADE(GRAD,RLN,CSN,RSG,RTP,EG)
C THIS SECTION OF SUBR GRADE READS A RESP. RECORD SEQUENTIALY
C AND GRADES THE TEST
      IMPLICIT INTEGER*2 (X)
      INTEGER FILES(3)/12,6,0,4,7/,IRESPT/4/,IQTPT/3/
      INTEGER*2 MORC(10,40),MR(65),RESPR(65),ANS(40),NUMQT(10),IP(5)
      INTEGER*2 GRAD,CSN,RSG,RTP,EG,LN,ST,CTGT,CM,RLN
      INTEGER*2 COUNT,IPE,GNO,RIGHT,ITEST,SG,SM
      INTEGER*2 SUBGRD
      EQUIVALENCE (RESPR(16),ANS(1)),(GNO,MR(13))
      EG=0
      GRAD=0
      COUNT=0
C READ RESPONSE TAPE
 3000 READ(IRESPT,END=9999)RESPR
      RLN=RESPR(1)
      IF(LN-RLN) 3020,3010,3000
 3010 CONTINUE
      CSN=RESPR(2)
      RSG=RESPR(3)
      RTP=RESPR(4)
C THIS EQUATION MAPS A 2-D FIELD INTO A 1-D FIELD
      IPE= RTP*2*(RSG-1)
      GNO=NUMQT(IPE)
C CALCULATE A NUMERIC GRADE FOR TEST
      DO 1000 K=1,GNO
C FUNCTION RIGHT CORRECTS ENG, QUESTIONS. (1=CORRECT,0=WRONG)
      ITEST= SUBGRD(RTP,MORC(IPE,K),ANS(K))
      COUNT=COUNT+ITEST
 1000 CONTINUE
      GRAD = COUNT/GNO
  999 RETURN
C EG=1 IS RETURN CODE FOR END OF FILE
 9999 EG=1
      RETURN
 3020 CONTINUE
      EG=-1
      LN=RLN
      BACKSPACE  IRESPT
      RETURN
C
C
      ENTRY MEDREI(LN,ST,IP,CTGT,CM)
      MR(13)=IQIPT,515)
C
C THIS SECTION PICKS OUT THE HEADERS REQUIRED TO GRADE THE TEST
C AND STORES THEM IN THE ARRAY CALLED  MORC
C
      DO 2010 K2=1,10
      NUMQT(K2)=1
      DO 2010 K3=1,40
      MORC(K2,K3)=0
 2010 CONTINUE
      CTGT=0
C THIS LOOP FETCHS HEADER RECORD AND STORES IT
C      PLUS COUNTS TOTAL EXAMS TO BE GRADED (CTGT)
C      AND NUMBER OF QUESTIONS PER TEST (NUMQT)
      DO 1500 KT=1,5
```

FILMED FROM BEST AVAILABLE COPY


```
      SC=1
      IF((IP(KI).LE.0.OR.IP(KI).GT.5) GO TO 1500
      IF(IP(KI).NE.4) GO TO 1530
 1520 DO 1540 SM=1,ST
      SC=SM
 1530 CALL GF(((MR,LN,SC,IP(KI),FILES,EM)
      IF(EM.NE.0) WRITE(IOTPT,510) EM,LN,IP(KI),SC
      IPE=IP(KI)+2*(SC-1)
      DO 1550 KI=1,46
      NUMC(IPE,KI)= MR(17+KI)
 1550 CONTINUE
      CTGT=CTGT+1
      NUMC(IPE)=CHD
      IF(IP(KI).NE.4) GO TO 1500
 1540 CONTINUE
 1500 CONTINUE
      RETURN
  510 FORMAT(' *** WARNING *** ERROR IN OBTAINING HEADER RECORD.  ERROR=
     1',I2,' LESSON=',I2,' TYPE=',I2,' SEGMENT=',I2)
  515 FORMAT(1H1)
      END
      SUBROUTINE REPIO
      IMPLICIT INTEGER*2 (K)
      INTEGER*2 SUBOUT(12),STDNME(13),CRNM(6),IP(5)
      INTEGER  OUTPUT(12),CSN
      INTEGER*2 ID(2,2),LID(2,2),LLN
      INTEGER*2 NP,LN,ST,SC
      INTEGER INPTC/1/,IOTPT/3/,IRESPT/4/,IAIMST/12/,IGRPT
      DATA ID(2,1)/'  '/,LID(2,1)/'  '/
      INPT=16
C READ IN THE REPORT REQUEST CARD
      READ(INPTC,10) NP,LN,ST,SC,(IP(K),K=1,5),(CRNM(K),K=1,6),IGRPT
C THIS BEGINS THE LOOP FOR THE OUTPUT OF GRADES
     1READ(IGRPT,521)
     1LN ,(STDNME(K),K=1,13),((ID(I,K),K=1,2),ID(2,2) , CSN,
     2(OUTPUT(K),SUBOUT(K),K=1,12)
 3000 CONTINUE
      LLN=LN
      DO 6000 J=1,2
      DO 6000 K=1,2
      LID(J,K)=ID(J,K)
 6000 CONTINUE
      BACKSPACE IGRPT
      IPGE=1
 5000 CONTINUE
      CALL HEADPG(IRPT,IPGE)
      WRITE(IOTPT,510)
      WRITE(IOTPT,515)(CRNM(K2),K2=1,6),LN
      WRITE(IOTPT,501)
      WRITE(IOTPT,530)
      DO 1000 JDUM3=1,45
C READ STUDENT BACKGROUND FILE
     1READ(IGRPT,521,END=999)
     1LN ,(STDNME(K),K=1,13),((ID(I,K),K=1,2),ID(2,2) , CSN,
     2(OUTPUT(K),SUBOUT(K),K=1,12)
      IF(LLN.NE.LN) GO TO 3000
      DO 4000 K=1,2
      IF(ID(SC,K).NE.LID(SC,K)) GO TO 3000
 4000 CONTINUE
```

FILMED FROM BEST AVAILABLE COPY

```
        AIMS III SOURCE STATEMENT LISTING

       LEN=LN
       DO 2000 J=1,2
       DO 2000 K=1,2
       LID(J,K)=ID(J,K)
 2000 CONTINUE
C OUTPUT NORMAL LINE OF STATISTICS
      WRITE(LUTP1,520)
     1     (STORMC(K),K=1,13),(ID(1,K),K=1,2),ID(2,2) , CSN,
     2(OUTPUT(K),SUBCOD(K),K=1,12)
 4000 CONTINUE
      GO TO 5000
   10 FORMAT(12,7X,312,314,9X,6A2,6A,12)
  501 FORMAT(1H0)
  510 FORMAT(/,50X,'*** VOLUME SUMMARY ***')
  515 FORMAT(//,40X,'COURSE ',6A2,5X,'VOLUME NO.  ',A2,/,20A,'NOTE
     1THE ** SIGNIFIES THAT NO MATERIALS HAVE BEEN SUBMITTED FOR PROCESS
     1ING')
  520 FORMAT(1X,12A2,A1,3A2,A3,12(2X,A4,A2))
  521 FORMAT(A2,12A2,A1,2A2,A2,A3,12(A4,A2))
  530FORMAT  (10A,'STUDENT',15X,'CSN  PRE-',3(3X,'STUDY'),
     1'  ASSIGN    HOME    POST   PERF.    CUR.   EAPE.',/,
     2 27A,'TEST   QUE(1)  QUE(2)  QUE(3)  QUE(4)  QUE(5)',
     3 11X,'NUMB    TEST    INDEX    AVE.    INDEX')
  540 FORMAT(1X,12A2,A1,6A,I5,6A,'STUDENT DROPPED')
  550 FORMAT(1X,12A2,A1,6A,I5,6X,'THIS STUDENT DOES NOT HAVE ANY MATERIA
     1LS TO BE PROCESSED')
  999 RETURN
      END
/*
// LKED.SYSLIB DD
// EXEC LNKEDT
/&

DOCID ES II ECP
HELP1 F
 END OF DATA

PAGE&DD
```